

**Monitoring a analýza kapacity  
dálničních parkovišť**

**Monitoring and Analysis of Highway  
Parking Capacity**

## Zadání diplomové práce

Student: **Bc. Jiří Novák**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Monitoring a analýza kapacity dálničních parkovišť**  
**Monitoring and Analysis of Highway Parking Capacity**

### Zásady pro vypracování:

V rámci diplomové práce se student zaměří na možnosti využití dat z profilových detektorů na dálničních parkovištích a dat z mýtných bran, které jsou relevantní k těmto parkovištím. Hlavním cílem, ke kterému tato data mohou sloužit je identifikace, zda je na parkovišti volno a to včetně krátkodobé predikce. Předtím, než je možno udělat samotnou predikci, je potřeba analyzovat možnosti dostupných dat pro tento problém – a to jak data z detektorů, tak geografický popis.

### Jednotlivé body práce jsou:

1. Nastudování relevantní dopravní terminologii vztahující se k zadání této práce.
2. Klasifikace parkovišť ve vztahu k mýtným branám a exitům.
3. Analýza vstupních dat – data z profilových detektorů, data z mýtných bran.
4. Definice možných výstupů, které bude možné na základě analýzy vstupních dat poskytnout uživatelům.
5. Vytvoření prototypové aplikace využívající data o obsazenosti parkovišť.
6. Testování a vyhodnocení dosažených výsledků.

### Seznam doporučené odborné literatury:

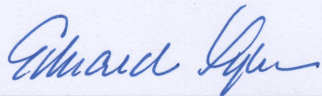
Todd Litman: Parking Management Strategies, Evaluation and Planning, Victoria Transport Policy Institute

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

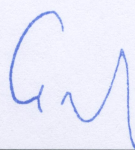
Vedoucí diplomové práce: **Ing. Jan Martinovič, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry

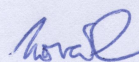


prof. RNDr. Václav Snášel, CSc.  
děkan fakulty



Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2014



.....

Rád bych na tomto místě poděkoval Mgr. Markovi Ščerbovi ze společnosti CDV za spolupráci při řešení této diplomové práce a za poskytnuté informace k problematice parkování. Dále bych také chtěl poděkovat Ing. Janu Martinovičovi, Ph.D. za vytrvalost při častých dotazech na problematiku mýtného systému a informací s tím spojených.



## Abstrakt

Práce se v teoretické části zabývá problematikou zákonů spojených s povinností přestávek v řízení a poskytování informací pro řidiče o volných parkovacích místech z důvodu parkování kvůli přestávek. Dále jsou zde probrány možnosti monitorování obsazenosti parkovišť, kde je jako jeden způsob uvažován mýtný systém. V praktické části je provedena nejprve klasifikace parkovišť na dálnici D1 vůči mýtným branám a také klasifikace sledovaných parkovišť s instalovanými senzory. Poté byla provedena analýza dat poskytovaných senzory a mýtnými branami a na základě této analýzy byly navrženy profily parkovišť a mýtných bran a byly definovány možné výstupy poskytované pro další zpracování. Nakonec byly vytvořeny a popsány dvě aplikace. První z nich slouží pro správu parkovišť a také poskytuje tyto informace vnějšímu světu. Druhou je simulátor, který umožňuje simulovat čas a vytvářet fiktivní vozidla pro mýtné brány nebo využívá ostrá data.

**Klíčová slova:** Mýtná brána, Wavetronix, obsazenost parkoviště, simulátor provozu

## Abstract

The work in theoretical part is related with problematic of the laws connected with obligation of breaks during driving and providing an information for a drivers about available parking place in case of break. There are also discussed options of monitoring of the availability of parking lots, where the system of toll gates is considered as one of the possibility. In the practical part of the work as first, a classification of the parking lots on the D1 highway is done and also a classification of parking lots with installed sensors. Then the analysis of data provided by toll gates and sensors is done and on the basis of this analysis were designed profiles of parking lots and toll gates, and outputs which can be provided for further processing are defined. At the end of the work the two applications are created and described. The first one serves for the managing of the parking lots and toll gates and also provides its data to the outside world. The second one is simulator which allows the time simulation and creates fictional vehicles for toll gates or it could use real data.

**Keywords:** Tollgate, Wavetronix, parking availability, trafic simulator

## Seznam použitých zkratek a symbolů

SPZ	– Státní poznávací značka
ŘSD	– Ředitelství silnic a dálnic
RDS-TMC	– technologie pro doručování dopravních informací řidičům dopravních prostředků (Radio data systém-Traffic Message Channel)
DSRC	– technologie pro komunikaci na krátkou vzdálenost (Dedicated short-range communication)
JSDI	– Jednotný systém dopravních informací
NDIC	– Národní dopravní informační centrum
OBU	– On-Board Unit – označení zařízení pro komunikaci s mýtnou bránou
IR	– infračervené zařízení
XML	– rozšiřitelný značkovací jazyk
CSV	– soubor s hodnotami oddělené čárkami (Comma-separated values)
GPS	– globální navigační systém (Global Positioning System)
WCF	– sada knihoven pro tvorbu servisně orientovaných aplikací (Windows Communication Foundation)
URL	– řetězec znaků pro identifikaci umístění zdrojů na internetu (Uniform Resource Locator)
GUI	– grafické rozhraní pro ovládání aplikace (Graphical user interface)



## Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
1.1	Struktura práce . . . . .	5
<b>2</b>	<b>Teoretický úvod do problematiky</b>	<b>7</b>
2.1	Zákony, nařízení a směrnice . . . . .	7
2.2	Jednotný systém dopravních informací . . . . .	8
2.3	Mýtný systém České republiky . . . . .	11
2.4	Monitorování obsazenosti parkovišť . . . . .	17
<b>3</b>	<b>Analýza a realizace monitorovací aplikace</b>	<b>21</b>
<b>4</b>	<b>Klasifikace parkovišť</b>	<b>22</b>
4.1	Parkoviště na dálnici D1 . . . . .	22
4.2	Mýtné brány, exity a parkoviště . . . . .	26
4.3	Sledovaná parkoviště se senzory . . . . .	27
<b>5</b>	<b>Obecná datová analýza vstupů</b>	<b>30</b>
5.1	Datová analýza mýtných bran . . . . .	30
5.2	Datová analýza snímačů na parkovištích . . . . .	30
<b>6</b>	<b>Poskytované výstupy</b>	<b>36</b>
6.1	Výstupy na základě mýtných bran . . . . .	36
6.2	Výstupy ze senzorů . . . . .	38
6.3	Doba strávená v úseku mezi branami . . . . .	38
6.4	Doba strávená v úseku mezi senzory . . . . .	39
6.5	Párování vozidel mezi branami a senzory . . . . .	39
6.6	Profily mýtných bran . . . . .	40
6.7	Profily parkovišť . . . . .	40
6.8	Profily parkovacích senzorů . . . . .	41
6.9	Predikce obsazenosti parkoviště . . . . .	41
6.10	Shrnutí možných poskytovaných výstupů . . . . .	42
<b>7</b>	<b>Implementace</b>	<b>44</b>
7.1	Parking Manager . . . . .	44
7.2	Gate simulator . . . . .	61
7.3	Získávání dat ze senzorů . . . . .	69
<b>8</b>	<b>Srovnání výsledků z mýtných bran a senzorů</b>	<b>71</b>
<b>9</b>	<b>Závěr</b>	<b>72</b>
<b>10</b>	<b>Reference</b>	<b>73</b>

## Seznam tabulek

1	Vhodnost použití snímací technologie . . . . .	19
2	Parkoviště na D1 ve směru Hulín → Praha . . . . .	23
3	Parkoviště na D1 ve směru Praha → Hulín . . . . .	24
4	Exity mezi branami a parkovišti Praha → Hulín . . . . .	28
5	Exity mezi branami a parkovišti Hulín → Praha . . . . .	28
6	Parkoviště s instalovanými snímači Wavetronix . . . . .	29
7	Příklad hodnot pro určení existujících vozidel . . . . .	31
8	Přepočtené hodnoty pro určení existujících vozidel . . . . .	32
9	Příklad hodnot pro určení překrývajících se vozidel . . . . .	33
10	Příklad hodnot pro spojení záznamů . . . . .	33
11	Příklad postupného spojení záznamů a výsledná délka . . . . .	34



## Seznam obrázků

1	Dopravní značení placených úseků (Zdroj [1]) . . . . .	11
2	Dopravní značení mýtného (Zdroj [4]) . . . . .	12
3	Přehled zpoplatněných úseků (zdroj [16]) . . . . .	12
4	Palubní zařízení Premid (zdroj [12]) . . . . .	13
5	Použitá zařízení na kontrolní mýtné bráně (zdroj [15]) . . . . .	14
6	Postup detekce vozidla kontrolní mýtnou bránou (zdroj [16]) . . . . .	15
7	Druhy mýtných bran v ČR (zdroj [16]) . . . . .	16
8	Rozmístění mýtných bran v ČR (zdroj [16]) . . . . .	16
9	Instalace snímače Wavetronix (zdroj [20]) . . . . .	20
10	Spojovací silnice mezi parkovišti v různých směrech (zdroj: maps.google.com)	25
11	Exit za příjezdovou bránou . . . . .	26
12	Exit před odjezdovou bránou . . . . .	26
13	Exity za příjezdovou a před odjezdovou bránou . . . . .	26
14	Exity před příjezdovou a za odjezdovou bránou . . . . .	26
15	Rozmístění a označení senzorů u parkovišť (zdroj: maps.google.com) . . .	27
16	Překrývající se vozidla . . . . .	32
17	Záznamy pro spojení . . . . .	33
18	Spojení dvou záznamů do jediného . . . . .	34
19	Třídní diagram pro parkoviště a mýtné brány . . . . .	44
20	Metoda ProcessVehicles . . . . .	48
21	Zpracování odjezdových vozidel v metodě ProcessVehicles . . . . .	49
22	Třídní diagram profilů bran a parkovišť . . . . .	51
23	Třídní diagram časových hodnot jednotlivých profilů . . . . .	53
24	Příklad aktualizace časového úseku . . . . .	56
25	Databázové tabulky pro uložení dat . . . . .	57
26	Třídní diagram Data mapperů pro práci s databází . . . . .	58
27	Třídní diagram webové služby pro poskytování dat . . . . .	60
28	Manuální nastavení profilů v simulátoru . . . . .	64
29	Třídní diagram propojení na Wavetronix databázi . . . . .	70

## Seznam výpisů zdrojového kódu

1	Metoda ProfileUpdate . . . . .	54
2	Určení referenčního času pro aktualizaci profilu parkoviště . . . . .	56
3	Aktivace jediné instance objektu pro webovou službu . . . . .	59
4	Přiřazení vytvořené instance objektu do hostu webové služby . . . . .	59
5	Vytvoření dvou endpointů pro JSON a XML formát . . . . .	60
6	Nastavení simulátorového času parkovišti . . . . .	62
7	Použití simulátorového času místo času aktuálního . . . . .	62
8	Cyklus volající kalkulační metody . . . . .	62
9	Řízení běhu simulátoru . . . . .	63
10	Využití manuálně nastavených hodnot profilů . . . . .	64
11	Náhodné určení počtů vozidel pro parkoviště a bránu . . . . .	65
12	Určení intervalu pro generovaná vozidla . . . . .	66
13	Generování vozidel s náhodným časovým rozestupem . . . . .	66
14	Přiřazení statusu vygenerovaným vozidlům . . . . .	66
15	Výběr vozidel pro zpracování příjezdovou bránou . . . . .	67
16	Výběr vozidel pro zpracování odjezdovou bránou . . . . .	68
17	Rozřazení dat do časových úseků . . . . .	69



## 1 Úvod

Problematika parkování nákladních vozidel je dnes hojně diskutované téma a to hlavně kvůli zákonům, které omezují délku řízení a nařizují povinné přestávky. Řidiči tak mají povinnost po určité době zastavit na delší dobu (řádově několik hodin) a odpočinout si, jinak by jim a také jejím zaměstnavatelům hrozila pokuta za nedodržení zákonů. V případě řízení po silnicích mimo dálnice se nejedná o tak velký problém. Jakmile se však řidič pohybuje na dálnici, je horší najít parkovací místo na přeplněných dálničních parkovištích a dojezd na další parkoviště může být z důvodů vypršení časové lhůty pro řízení také značný problém.

Na základě tohoto faktu vydala Evropská komise několik nařízení a doporučení, která by měla zajistit vybudování systému pro sledování obsazenosti parkovišť za účelem poskytovat informace o volných parkovacích místech. Tyto informace by měly sloužit řidičům pro plánování, kdy a kde se zastaví, aby si odpočinuli.

Pro monitorování parkovišť je možné využít různé technologie, které budou nastíněny také v této práci. Žádná ale není momentálně na parkovištích instalovaná. Proto se tato práce bude zabývat prozkoumáním možnost monitorování obsazenosti dálničních parkovišť pomocí již zaběhnutého mýtného systému.

Aby bylo možné sledovat pohyb nákladních vozidel na parkovištích, musí být zajištěna dostatečná hustota snímačů (ve zkoumaném případě mýtných bran). Proto v práci popíšeme analýzu mýtného systému, klasifikaci parkovišť vůči mýtným branám a také faktory, které by mohly mít vliv na přesnost určování obsazenosti.

Pro řešení práce byla k dispozici data ze dvou parkovišť s instalovanými zvláštními senzory pro monitoring parkovišť jako testovací provoz. Bude se tedy hovořit také o analýze dat, které poskytují mýtné brány ve srovnání s daty, které jsou k dispozici z těchto senzorů. Následovat bude vytýčení dat, které bude možné poskytovat řidičům a dalším subjektům právě z mýtného systému o obsazenosti parkovišť.

Cílem implementační části práce a fáze testování bylo vytvořit software pro zpracování dat z mýtného systému. Nejprve bude popsán software, který má za úkol správu informací o parkovišti a šíření dat z nich do okolního světa a pak bude následovat popis simulátoru, který má za úkol jak simulaci času s využitím reálných dat, tak i vytváření fiktivních vozidel.

Na konci této práce budou srovnány výstupy z obou těchto aplikací s daty, které poskytují snímače instalované na dvou testovacích parkovištích. Tyto dvě parkoviště jsou brány jako srovnávací a na základě dat z nich se pak určí, jak přesné je využití mýtného systému pro monitorování parkovišť.

### 1.1 Struktura práce

Začátek práce se v kapitole 2 věnuje nastínění problematiky zákonů a nařízení (2.1), popisem systému (2.2), který by do budoucna měl shromažďovat a poskytovat informace a popisem mýtného systému (2.3) společně s dalšími možnostmi monitorování parkovišť.

Kapitola 3 uvozuje samotné řešení celé práce, které se skládá z prozkoumání parkovišť na dálnici D1 (4.1), dále jejich klasifikace vůči mýtným branám (4.2) a také uvedení

dvou sledovaných parkovišť na dálnici D5 s instalovanými senzory (4.3). Analýza dat, které jsou k dispozici z mýtných bran a senzorů je rozebrána v kapitole 5. Následně jsou probrány poskytované výstupy (6), které bude možno poskytovat řidičům případně dalším subjektům. Jedná se hlavně o dobu, jakou vozidlo strávilo v úseku (6.3) a o časové profily parkovišť a bran (6.6 a 6.7). Předposlední kapitola 7 popisuje postupně aplikaci pro správu dat z parkovišť (7.1) a simulátor provozu na mýtných branách a parkovištích (7.2). Všechny výsledky a srovnání výstupů z mýtných bran a senzorů jsou uvedeny v kapitole 8.

## 2 Teoretický úvod do problematiky

Celá práce se zabývá problematikou povinných přestávek profesionálních řidičů. S tímto je spojená také problematika hledání parkovacího místa v době, kdy již musí řidič začít svůj odpočinek. Doba, po jakou může řidič z povolání nejdéle řídit a naopak doba, kterou musí nejméně odpočívat, je zakotvená v českém zákoníku a také v nařízení evropské unie. Nejprve jsou tedy stručně uvedeny všechny zákony s touto problematikou spojené.

Dále je také v zákonech a nařízeních specifikována budoucí povinnost poskytovat řidičům informaci o volných parkovacích místech v okruhu, kde je možné bezpečně dojet, aniž by překročili nejdelší zákonnou dobu řízení. Tato informace by měla být v budoucnu poskytována jedním ze systémů pro sběr a poskytování informací, které jsou v České republice již nyní k dispozici.

Aby však mohly být tyto informace poskytovány, musejí být nejprve nějakým způsobem shromážděny. Protože je aktuálně největší problém s parkovacími místy pro řidiče z povolání (hlavně pak řidiče nákladních vozidel) na dálnicích, je další zpracování vedeno hlavně na systém dálnic v České republice a na parkoviště na těchto dálnicích.

Monitorování obsazenosti parkovišť pak je možné provést několika způsoby a to buď instalací nových snímacích prvků, nebo využití již instalovaných technologií. V případě instalace nových technologií je možné použít několik základních technik monitorování průjezdnosti, jak bude dále uvedeno. Co se týče využití již instalovaných zařízení, uvažuje se o využití stávajícího mýtného systému pro monitorování obsazenosti parkovišť na základě časových dat.

Všechny tyto problémy budou blíže probrány v následujících kapitolách.

### 2.1 Zákony, nařízení a směrnice

#### 2.1.1 Doba odpočinku a povinné přestávky

Na základě Nařízení Evropského parlamentu a rady (ES) č. 561/2006 ze dne 15. března 2006 [7] mají řidiči nákladních vozidel, jejichž maximální přípustná hmotnost přesahuje 7,5 tun, povinnost dodržovat povinné *přestávky v řízení* během kterých nesmí řidič řídit ani provádět jinou práci. *Jinou prací* je myšlena činnost jako řízení a také jakákoli práce pro téhož nebo jiného zaměstnavatele v odvětví dopravy či jinde.

Po čtyřech a půl hodinách řízení musí mít řidič nepřerušenu přestávku nejméně 45 minut, pokud mu nezačíná doba odpočinku.

Tato přestávka může být nahrazena přestávkou v délce nejméně 15 minut, po níž následuje přestávka v délce nejméně 30 minut, které jsou v období rozloženy tak, aby byla v souladu s předchozím odstavcem.

Dále musí řidič dodržovat povinnou *denní dobu odpočinku*, která je definována jako denní doba, během níž může řidič volně nakládat se svým časem a která zahrnuje *běžnou denní dobu odpočinku* nebo *zkrácenou denní dobu odpočinku*.

*Běžnou denní dobu odpočinku* se rozumí doba odpočinku v celkovém trvání nejméně 11 hodin. Tuto běžnou dobu odpočinku lze případně rozdělit do dvou časových úseků,

z nichž první musí být nepřerušená doba v celkovém trvání nejméně 3 hodin a druhý nepřerušená doba v celkovém trvání nejméně 9 hodin.

*Zkrácenou denní dobou odpočinku* se rozumí doba odpočinku v celkovém trvání nejméně 9 hodin, ale kratší než 11 hodin.

*Denní dobou řízení* se myslí celková doba řízení mezi skončením jedné denní doby odpočinku a začátkem druhé denní doby odpočinku nebo mezi denní dobou odpočinku a týdenní dobou odpočinku.

Denní doba řízení nesmí přesáhnout 9 hodin.

V průběhu každých 24 hodin po skončení předchozí denní nebo týdenní doby odpočinku musí mít řidič novou denní dobu odpočinku.

### 2.1.2 Rozšíření inteligentního dopravního systému

Problém se zahájením odpočinku nastává u dálnic a to zejména v delších úsecích mezi jednotlivými sjezdy. V takovém případě může nastat situace, kdy řidiči může vypršet časový limit pro povolenou dobu řízení dřív, než dojde povolenou rychlostí k nejbližšímu odstavnému parkovišti nebo sjezdu, kde může dále snadno zaparkovat, aby zahájil dobu odpočinku.

Pro tento účel vydala Evropská komise Nařízení komise v přenesené pravomoci (EU) č. 885/2013 ze dne 15. května 2013 [2], kterým se doplňuje směrnice Evropského parlamentu a Rady 2010/40/EU o inteligentních dopravních systémech, pokud jde o poskytování informačních služeb týkajících se bezpečných a chráněných parkovacích míst pro nákladní a užitková vozidla.

Toto nařízení v případě České republiky nerozšiřuje pouze předchozí směrnici Evropské komise, ale také přímo zákon 13/1997 Sb. [8] část osmá o inteligentních dopravních systémech §39a odstavec 1: Inteligentní dopravní systém je souborem elektronických prostředků, technických zařízení, programového vybavení a jiných nástrojů, které umožňují vyhledávání, shromažďování, zpřístupňování, používání a jiné zpracovávání údajů o pozemních komunikacích, silničním provozu, cestování, logistice a dopravním spojení, a jehož účelem je zvýšení bezpečného a koordinovaného užívání pozemních komunikací a snížení negativních dopadu silničního provozu na životní prostředí.

Rozšíření hovoří o *bezpečném a chráněném parkovacím místě* jako o parkovacím místě pro komerční uživatele, jež jim umožní vyhnout se nevhodnému parkování a přispěje k bezpečnosti řidičů a nákladu. Takové místo je mimo jiné také využíváno pro parkování v případě, kdy řidič potřebuje zahájit povinnou dobu odpočinku.

Dále je díky tomuto rozšíření specifikováno, že řidiči musejí mít informace k dispozici prostřednictvím vhodného komunikačního kanálu, aby je mohli dynamicky využívat a přizpůsobit naplánování další cesty.

## 2.2 Jednotný systém dopravních informací

Jednotný systém dopravních informací (JSDI) je projekt pro Českou republiku, který byl schválen již v roce 2005 usnesením vlády č. 590 a je to tedy státem podporovaný projekt, na jehož realizaci se podílí ve spolupráci Ministerstvo vnitra, Ministerstvo informatiky,

Ministerstvo dopravy, Ředitelstvím silnic a dálnic ČR a řada dalších orgánů a institucí veřejné správy.

Cílem je sjednotit roztržštěné dopravní zpravodajství od různých subjektů jako jsou např. policie ČR, Hasičský záchranný sbor a další, do jediného centralizovaného systému. Tento systém by sloužil také pro subjekty, který podobný systém neměly nebo byl také decentralizovaný (např. zřízení Centrální evidence komunikací).

### 2.2.1 Poskytovatelé informací pro JSDI

Podle novely zákona č. 361/2000 Sb. [9], o provozu na pozemních komunikacích, je nově stanovena povinnost pro Ministerstvo dopravy nebo jím pověřený subjekt, aby informoval o mimořádných informacích na pozemních komunikacích, které jakýmkoli způsobem ovlivňují bezpečnost nebo plynulost provozu. Tato novela také udává povinnost pro Policii ČR, obecní policii, silniční a správní úřady správce pozemních komunikací (Ředitelství silnic a dálnic, Technické správy komunikací velkých měst, krajské a obecní úřady) a Hasičský záchranný sbor, poskytovat tomuto subjektu aktuální informace.

Do budoucna se počítá také se zapojením dalších informačních kanálů a to např. od podniků Povodí a Vodoprávních úřadů, systému dopravních zpravodajů a mimo jiné také informací z elektronického mýtného, které je schopno na základě sesbíraných informací určovat hustotu a rychlost pohybu nákladních vozidel.

### 2.2.2 Obsah JSDI

Podle Ministerstva vnitra by JSDI měl obsahovat nebo již obsahuje tyto informace (převzato z [11]): „informace o okamžité hustotě a rychlosti dopravního proudu v daném úseku vybrané pozemní komunikace, informace o dopravních nehodách, o požáru vozidel na komunikaci nebo v jejím bezprostřední blízkosti, o požáru objektů v blízkosti komunikace, pokud mohou ohrozit provoz na komunikaci, informace o uzavírkách a objíždkách, o zvláštním užívání komunikace, pokud dojde k omezení průjezdnosti pozemní komunikace nebo pokud se po pozemní komunikaci pohybuje zvýšené množství chodců, o jiné překážce provozu (např. vysypaném nákladu, nepojízdném vozidle), informace o opravách a údržbě komunikace, o meteorologické situaci a povětrnostních podmínkách, které mají vliv na průjezdnost komunikace, sjízdnosti komunikace, o haváriích (např. vodovodních) sítí v tělese komunikace nebo bezprostřední blízkosti komunikace, o nařízeném aktuálním omezení průjezdnosti komunikace pro určité typy vozidel, o pohybu nadměrných a nebezpečných nákladů, informace z dopravně telematických aplikací (jako jsou např. proměnné informační tabule na D1), informace o aktuálních průtocích na sledovaných tocích, které by mohly při povodních ohrozit provoz na pozemních komunikacích, poruchách světelné signalizace, informace o obsazenosti záchytných parkovišť, jiném nebezpečí způsobeném nepředvídanou nebo krizovou situací, atd.“

Všechny tyto informace jsou nebo budou uloženy v centrálním datovém skladu, který je v provozu od roku 2006 a jehož poskytovatelem je ŘSD ČR. Veškeré informace v tomto skladu jsou ověřovány a autorizovány operátory v Národního dopravního informačního centra (NDIC), které se nachází v Ostravě.



Informace z JSDI jsou dostupné v datovém formátu XML prostřednictvím standardních webových služeb na základě jednotné typové smlouvy a je možné je odebírat zdarma. „Data může odebírat každá právnická nebo fyzická osoba, která zajistí jejich další efektivní šíření ve prospěch zvýšení bezpečnosti a plynulosti silničního provozu nebo která je bude využívat ve svých informačních systémech na podporu vlastních procesů a aktivit souvisejících s dopravou a provozem na pozemních komunikacích.“ [18]

### 2.2.3 Národní dopravní informační centrum

NDIC provozuje na základě rozhodnutí Vlády ČR č. 590 ze dne 18. 5. 2005 [10] a v souladu s §124 odst. 3 zákona č. 361/2000. Sb. [9] ve znění pozdějších předpisů Ředitelství silnic a dálnic ČR.

NDIC zajišťuje tyto služby [17]:

- kontroluje kvalitu a správnost předávaných dopravních informací a dopravních dat,
- kontroluje předávání informací od jednotlivých zapojených orgánů, organizací, institucí, osob a subjektů, a v případě neplnění jejich povinností nebo metodického postupu problém řeší,
- řeší konflikty v případě, že do systému přijdou o jedné události ve stejném časovém intervalu ze stejného místa, úseku nebo oblasti stejné nebo podobné dopravní informace a odpovídá za vydání sjednocené informace o této události,
- ve spolupráci s příslušnými orgány, organizacemi a institucemi provádí aktualizaci informací o události v závislosti na konkrétním vývoji situace v místě,
- sleduje životní cyklus vývoje událostí až do jejich ukončení a plného obnovení provozu,
- neautorizované nebo neúplné informace doplňuje o další atributy z dalších zdrojů a provádí jejich ověření,
- poskytuje dopravní informace a dopravní data všem odběratelům, zajišťuje provoz systémů pro publikaci a distribuci dopravních informací a dopravních dat,
- provozuje dílčí aplikace a systémy, řeší případné technické a technologické problémy,
- vede a spravuje archiv historických dat a dopravních informací, atd.

### 2.2.4 Získávání informací z JSDI

Informace, které může řidič získat z JSDI, se dělí na dva typy:

- informace získané před jízdou,
- informace poskytované v reálném čase během jízdy.

Informace získané před jízdou je možné obdržet na webových stránkách <http://www.dopravniinfo.cz>, kde jsou k dispozici informace o:

- nehodách
- sjízdnosti komunikací
- stupních provozu
- nehodových místech.

Informace během jízdy je možné získat několika způsoby

- prostřednictvím rozhlasových stanic
- na call centrech některých komerčních subjektů jako je ÚAMK nebo Global assistance
- ze zařízení RDS-TMC, které přijímají aktuální informace přes FM rádiový signál a zobrazují je v mapě navigačního zařízení.

## 2.3 Mýtný systém České republiky

### 2.3.1 Zpoplatněné úseky dopravních komunikací

Na území české republiky jsou dopravní komunikace děleny do dvou kategorií – placené a neplacené. Zpoplatněné jsou všechny dálnice (vyjma některých úseků vyhrazených např. pro objížďky v případě výkopových prací), rychlostní silnice a některé části vybraných silnic I. třídy.

Za placené komunikace se odvádí poplatky dvěma možnými způsoby. Mezi první způsob patří časový kupon, tzv. dálniční známka, který je nalepený na čelním skle vozidla a jeho platnost je omezena na dobu určitou od jeho zakoupení (např. jeden měsíc) nebo na dobu 14 měsíců u ročních kuponů. Tuto variantu uhrazení poplatku je nutné použít na všech komunikacích označených dopravními značkami dálnice nebo silnice pro motorová vozidla 1, ale pouze pro vozidla, jejichž povolená hmotnost nepřesahuje 3,5 tuny.



(a) Dálnice



(b) Silnice pro motorová vozidla

Obrázek 1: Dopravní značení placených úseků (Zdroj [1])

Vozidla s celkovou hmotností přesahující 3,5 tuny jsou povinna na těchto silnicích hradit poplatky přes tzv. elektronický mýtný systém. Úseky, kde se hradí poplatky tímto způsobem, jsou označeny dopravními značkami dálnice nebo silnice pro motorová vozidla. Pro vozidla s touto hmotností jsou však mimo tyto dva typy komunikací zpoplatněny ještě další úseky silnic I. třídy a ty se označují dopravní značkou mýtné 2.



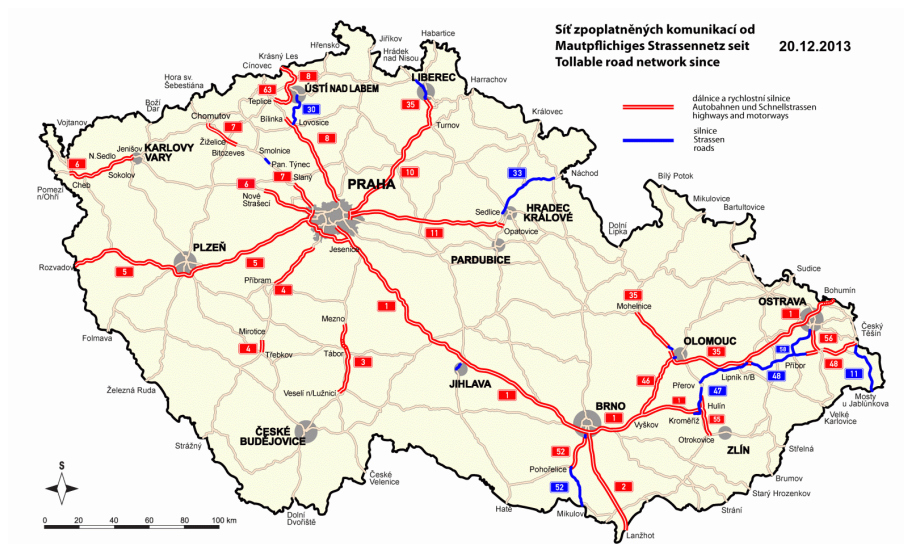
(a) Začátek mýtného



(b) Konec mýtného

Obrázek 2: Dopravní značení mýtného (Zdroj [4])

Veškeré úseky komunikací zpoplatněné elektronickým mýtným systémem jsou uvedeny v přehledové mapce 2.



Obrázek 3: Přehled zpoplatněných úseků (zdroj [16])

### 2.3.2 Způsob realizace elektronického mýtného

Elektronické mýtné je možné realizovat dnes již několika způsoby [6]:

- Satelitní snímání – vozidla v sobě mají instalována zařízení, které je možné sledovat pomocí satelitu a tak určovat jeho polohu a na základě pohybu po silnicích zjistit, zda se pohybuje po placeném úseku či nikoli. Pro tento způsob je nutné mít ve vozidle instalované zařízení pro komunikaci se satelitem.

- Průjezdni mýtná brána se závorou – na silnici je instalována průjezdni mýtná závor, u které je řidič povinen zastavit a zaplatit a až poté může pokračovat dále po placeném úseku silnice. Takovéto mýtné závory jsou většinou instalovány na začátku a na konci placeného úseku. Výhodou řešení pomocí průjezdních mýtných závor je, že odpadá instalace dalšího zařízení do projíždějících vozidel. Nevýhodou je však snížení plynulosti provozu na silnici.
- Průjezdni mýtná brána bez závor – vozidlo projíždějící mýtnou bránou nemusí zastavovat. Je ale potřeba mít ve vozidle instalované zařízení pro komunikaci s mýtnou bránou. Toto zařízení se při průjezdu mýtnou bránou automaticky spojí s čtečkou umístěnou na bráně a dojde k automatickému odečtení kreditu. Nesnižuje se tak plynulost provozu a řidiči nemusí řešit placení.

V České republice je použit třetí způsob realizace – průjezdni mýtná brána bez závor. To ve výsledku znamená, že všechny vozidla, která jsou povinna platit mýtné, musí být vybavena elektronickým zařízením pro komunikaci s mýtnou bránou. Jedná se o DSRC zařízení pro komunikaci na krátkou vzdálenost [14] s označením Premid 4, které obstarává komunikaci při průjezdu a na kterém je uložena také informace o množství kreditu. Každé takovéto zařízení má jednoznačný OBU [5] identifikátor, který se používá pro pozdější identifikaci majitele.

Jakmile vozidlo projede bránou, je řidič upozorněn akustickým signálem, zda transakce odečtu mýtného proběhl úspěšně či nikoli. Premid zařízení je také schopno upozornit řidiče na nízkou výši kreditu a upozornit jej, že již nemá dostatečné finanční prostředky na to, aby dále pokračoval po zpoplatněném úseku a aby jej na nejbližším sjezdu opustil.

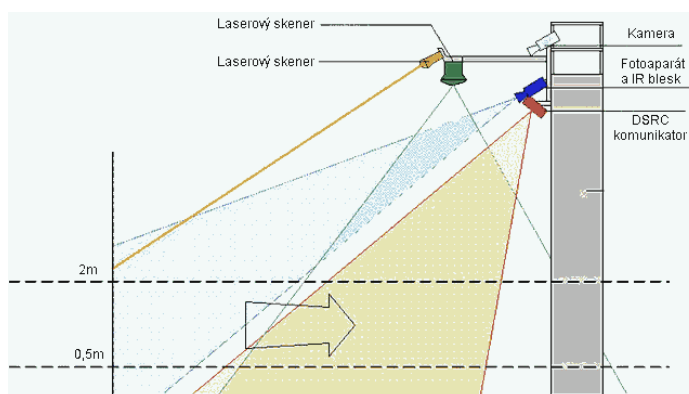


Obrázek 4: Palubní zařízení Premid (zdroj [12])

Samotná mýtná brána je konstrukce nad silnicí, která na sobě nese několik zařízení pro každý jízdní pruh, nad kterými je instalována:

- Laserový skener 1 – zajišťuje detekci vozidla vjíždějícího do zorného pole mýtné brány.
- Laserový skener 2 – stará se určení délky projíždějícího vozidla a na základě určené délky provede klasifikaci do určité kategorie.
- Kamera – zaznamenává průjezd vozidla pro případné dodatečné manuální zpracování v centrále.

- Fotoaparát s IR bleskem – pořizuje fotku přední části vozidla. Na této fotce je mimo jiné také zaznamenána SPZ vozidla.
- DSRC komunikátor – zajišťuje samotnou komunikaci brány se zařízením nainstalovaným ve vozidle.



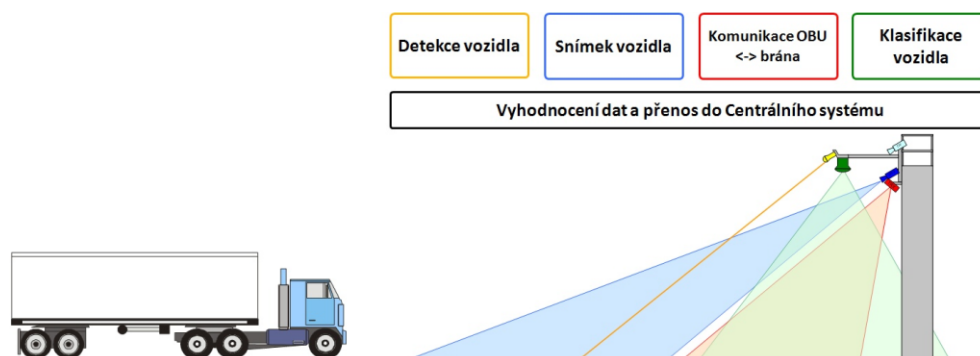
Obrázek 5: Použitá zařízení na kontrolní mýtné bráně (zdroj [15])

Zpracování projíždějícího vozidla probíhá následujícím způsobem:

1. Vozidlo je zaznamenáno prvním laserovým skenerem.
2. Vyfotí se čelní část vozidla včetně SPZ.
3. DSRC komunikátor se spojí s palubním zařízením Premid, odečte kredit za průjezd daným zpoplatněným úsekem a uloží si jeho OBU identifikační číslo.
4. Druhý laserový skener změří délku vozidla a provede klasifikaci.
5. Celý průběh je natočen kamerou pro případné ruční zpracování.
6. Na základě nasnímaných údajů dojde ke spojení majitele Premid zařízení s vozidlem podle SPZ (je automaticky rozpoznána z vyfoceného snímku) a z délky vozidla je odvozeno, zda se jedná o vozidlo, pro které je kredit předplacený či nikoli.
7. V případě, že se vozidlo neshoduje s daty v databázi, je automaticky odesláno upozornění na dispečink a tam se provede dodatečná analýza na základě videozáznamu.

Mýtné brány jsou také schopné detekovat vozidla, které na palubě nemají instalováno Premid zařízení. V případě, že z nasnímaných údajů systém vyhodnotí, že se jedná o vozidlo, které Premid zařízení musí mít, je opět automaticky upozorněn dispečink na protiprávní jednání. Tento systém eliminuje možné podvody s odstíněním zařízení ve vozidle tak, aby nedocházelo k odčítání kreditu.





Obrázek 6: Postup detekce vozidla kontrolní mýtnou bránou (zdroj [16])

### 2.3.3 Druhy mýtných bran v ČR

Na všech českých silnicích, které spadají do zpoplatněného mýtného systému, jsou instalovány dva druhy mýtných bran:

- Kontrolní mýtná brána – jedná se o bránu, která je popsána výše, tedy o kovovou konstrukci vedenou napříč nad monitorovanou silnicí, která nese pět zařízení – dva laserové skenery, kameru, fotoaparát s IR bleskem a DSRC komunikátor pro komunikaci s palubním zařízením Premid.
- Standardní mýtná brána – konstrukce je podobná jako u kontrolní mýtné brány, ale instalovaných zařízení není tolik. Na této mýtné bráně jsou instalovány pouze DSRC komunikátory a slouží pouze pro odečet kreditu z palubní jednotky.

### 2.3.4 Poloha bran

V České republice je momentálně vystavěný mýtný systém, který zpoplatňuje přibližně 1300km silnic. O sledování a odečítání poplatků za mýtné se stará okolo 250 mýtných bran, ale vzhledem k tomu, že se jedná o brány, které bývají často napříč celou silnicí, tedy přes oba jízdní směry, může se uvádět počet dvojnásobný, tedy přibližně 500 mýtných bran na celou síť zpoplatněných silnic. Všechny tyto brány komunikují v reálném čase s hlavním řídicím střediskem.

Rozložení standardních a kontrolních mýtných bran zvlášť není zmapováno, ale je k dispozici mapa rozložení všech mýtných bran v ČR včetně jejich seznamu, na kterých úsecích dálnic a silnic se nacházejí 8.

Mýtné brány jsou umístěny vždy mezi dvěma exity (sjezdy a nájezdy z dálnice). Tak je docíleno, že je zpoplatněn celý úsek, protože řidič nemá možnost z monitorované silnice sjet nikde jinde a musí tak pokaždé projet pod mýtnou bránou.

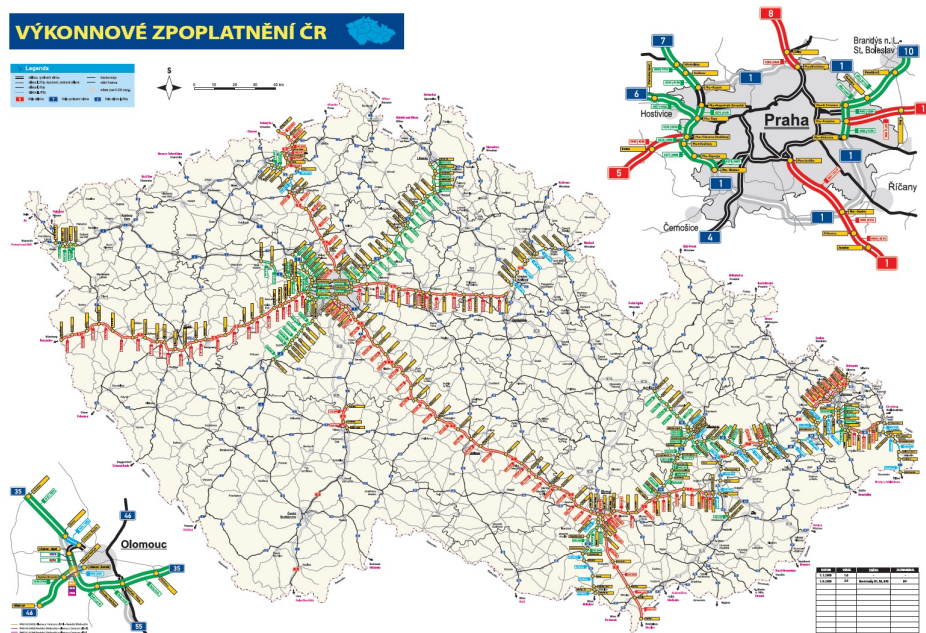


(a) Standardní mýtná brána



(b) Kontrolní mýtná brána

Obrázek 7: Druhy mýtných bran v ČR (zdroj [16])



Obrázek 8: Rozmístění mýtných bran v ČR (zdroj [16])

## 2.4 Monitorování obsazenosti parkovišť

Pro monitorování obsazenosti parkovišť je možné využít několik různých technik v závislosti na umístění, rozloze a frekventovanosti parkoviště. V zásadě se tyto techniky dají rozdělit do dvou kategorií:

- Monitorování každého parkovacího místa zvlášť
- Monitorování parkoviště jako celku

Přičemž každá z těchto technik má své pro a proti a lze ji využít při různých situacích. Nakonec této kapitoly bude uvedena jedna z technologií pro monitorování průjezdnosti vozidel. Tato technologie byla dále použita na kontrolních parkovištích pro sledování počtu vozidel na parkovišti. Jedná se o technologii Wavetronix

### 2.4.1 Monitorování každého místa zvlášť

Tato technika se využívá především u parkovišť, kde je přesně stanoveno, kde je možné zaparkovat. V případě, že řidič zastaví na místě jiném, než je pro parkování vyhrazeno, je okamžitě pokutován nebo musí parkovací prostor okamžitě opustit.

Parkovací místa, která jsou pevně vymezená, se mohou nacházet jak v uzavřeném prostoru (podzemní parkoviště nebo parkoviště u nákupních center) nebo parkoviště na otevřeném prostranství (parkování v ulicích s vyhrazenými místy pro stání).

Pro monitorování takovýchto parkovacích míst je dále možné využít několik různých aplikací, které snímají obsazenost daného místa:

- Čidlo parkovacího místa umístěné v zemi
- Čidlo parkovacího místa umístěné nad zemí
- Kamerový systém s rozpoznáváním obrazu pro detekci obsazenosti parkovacího místa

Ne všechny z těchto aplikací lze využít na všech parkovištích a to z mnoha důvodů. Například snímače umístěné nad zemí se nejčastěji využívají v podzemních garážích. Důvodem je jednoduchost jejich instalace, kdy není potřeba instalovat žádnou další konstrukci, která by snímač nesla. Takovéto čidlo by sice mohlo být instalováno i na parkovišti v otevřeném prostoru, kde není žádný strop nebo střecha, ale bylo by potřeba pro každé parkovací místo dále instalovat konstrukci držící snímač nad parkovacím místem.

Pro čidla umístěná v zemi už tolik omezení není. Je však potřeba zvážit jaký typ tohoto čidla použít. Na trhu jsou dnes k dostání dva typy, které se liší způsobem připojení k centrální jednotce:

- Čidlo bezdrátové – tato čidla jsou samostatná, pouze navrtaná v zemi bez nutnosti přivádění další kabeláže
- Čidlo drátová – čidlo je k centrální jednotce připojeno kabelem

U drátových čidel je výhoda absence napájecího zdroje, které může být součástí centrální jednotky a tato je přímo připojená ke zdroji napájení. Nevýhodou je však potřeba narušit vozovku nebo podlahu parkoviště a v případě venkovní instalace může v takovýchto případech v zimních měsících docházet kvůli povětrnostním vlivům k poškození narušené vozovky zmrzlou vodou zateklou do špatně utěsněné drážky pro kabeláž.

Další nevýhodou kabelových čidel je omezení vzdálenosti od centrální jednotky. Čím dále je centrální jednotka od čidla, tím delší kabel musí být a tím jsou větší ztráty způsobené odporem kabelu a také větší náklady na umístění kabeláže do podlahy nebo vozovky.

U bezdrátových čidel odpadá nutnost poškozovat vozovku z důvodu vedení kabeláže. Je však nutné zajistit napájení čidel akumulátorem, ty se však musejí v pravidelných intervalech měnit. Navíc přenos dat mezi čidlem a centrální jednotkou může být snadněji rušen. Vzdálenost od centrální jednotky již není tak limitující, ale také není neomezená.

Třetím způsobem je kamerový systém. Jedná se o kameru umístěnou tak, aby zabírala několik parkovacích míst najednou. Na základě rozpoznávání obrazu se detekuje, zda jsou parkovací místa volná či obsazená. U tohoto řešení odpadá nutnost jakýmkoli způsobem narušovat povrch parkovací plochy. Avšak tato metoda je takřka nepoužitelná v malých prostorech (např. podzemních parkovištích), kdy je kamera schopná zabrat pouze malý radius několika parkovacích míst a na pokrytí celého parkoviště by bylo třeba velkého množství kamer, které vyjdou ve výsledku draž než nadzemní nebo podzemní čidla.

Kamerový systém je účinný u parkovišť na otevřeném prostranství, kde je jedna kamera schopná monitorovat velké množství parkovacích míst. Pokud je však kamera použita venku, může docházet ke zkreslení výsledků kvůli zhoršené viditelnosti (např. silná mlha), proto ani tento způsob není 100% účinný.

#### **2.4.2 Monitorování parkoviště jako celku**

Pro parkoviště, kde je omezený vjezd a výjezd, tedy je zde například pouze jeden vjezd a jeden výjezd, je výhodnější využít monitorování této oblasti jako celku než každé místo zvlášť. Takovéto plochy je pak možné monitorovat pouze na základě počtu vozidel, které přijely nebo odjely jedním z několika průjezdních míst.

Tento způsob monitorování je se provádí prakticky jediným způsobem – průjezdní závorou. Průjezdní závora ale může mít několik podob. Mezi základní typy patří závora mechanická, u které musí řidič zastavit a nějakým způsobem si závoru otevřít (přístupovou kartou, vytištěním parkovacího lístku apod.). Druhý typ závory je závora elektromechanická. U té řidič zastavovat nemusí a jeho přítomnost je zaznamenána na základě např. přerušení paprsku nebo detekce mikrovlnným radarem. Existují však ještě jiné způsoby pro tento účel.

Všechny výše uvedené způsoby jsou uvedeny v přehledové tabulce 1.

Typ snímání	Druh parkoviště		
	Venkovní s více příjezdy a odjezdy	Venkovní s jedním příjezdem a jedním odjezdem	Podzemní
Čidlo v zemi	Ano*	Ano*	Ano
Čidlo nad parkovacím místem	Ano*	Ano*	Ano
Kamerový systém	Ano	Ano	Ne
Mechanická závora	Ne	Ano	Ano**
Elektronická závora	Ne	Ano	Ano

\* - pouze pokud jsou parkovací místa pevně vyhrazena a parkování mimo takovýchto míst je zakázáno

\*\* - používá se jako doplňující informace pro zamezení vjezdu - např. zbývá již pouze jedno místo, ale řidič na něj ještě nedorazil, takže se tváří jako volné, ale podle počtu vozidel na parkovišti je celé parkoviště plně obsazené

Tabulka 1: Vhodnost použití snímací technologie

### 2.4.3 Technologie Wavetronix

Pro realizaci dříve popsaných způsobů monitorování je možné použít nejrozličnější technologie od celé řady výrobců a to jak zahraničních, tak i tuzemských.

Jednou variantou jsou senzory společnosti Wavetronix [19], které využívají mikrovlnnou technologii ke snímání pohybu před senzorem. Není potřeba, aby vozidla při průjezdu zastavovala a jedná se tedy o elektronickou závoru. Tento typ senzoru je namontován na stožáru vedle silnice, kterou má snímat. Doporučená výška, do které má být zařízení instalováno je přibližně 2,7 – 15m. Protože snímač nesnímá pouze jeden jízdní pruh, ale je možné, aby snímá až 22 pruhů, je doporučeno instalovat senzor do větší výšky, aby nedocházelo k zastínění ostatních pruhů vozidly, která jsou blíže snímači. Dále je třeba dodržet minimální vzdálenost senzoru od snímané vozovky, tato vzdálenost je přibližně 1,83m.

V případě využití na parkovištích u dálnic je možné počítat, že na tyto parkoviště je pouze jedna příjezdová a jedna odjezdová cesta. Využijí se tedy pouze dvě zařízení – jedno na příjezdu a jedno na výjezdu. Tím je pokryto celé parkoviště a počítá se tak, kolik vozidel na parkoviště přijelo a kolik z něj odjelo a rozdíl těchto dvou hodnot dá celkový počet vozidel, který se v daném čase na parkovišti nachází.

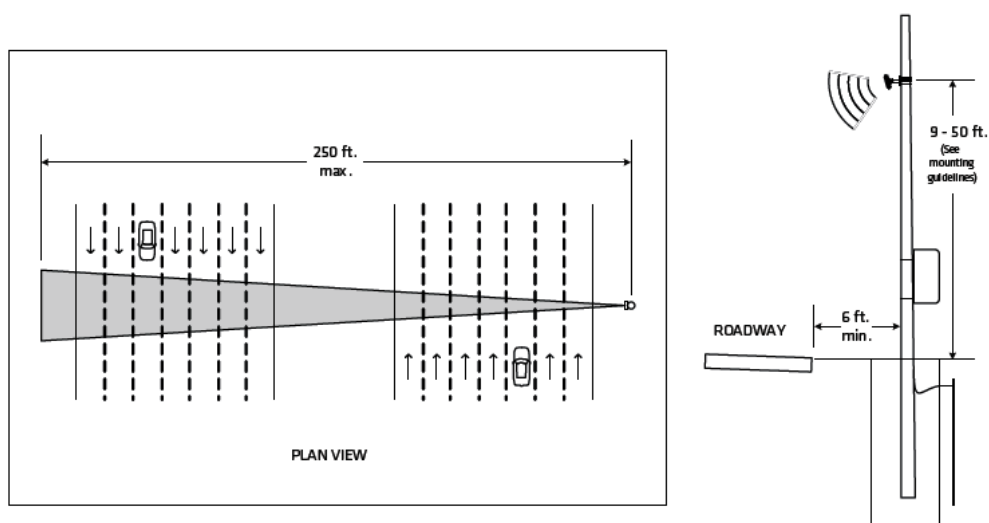
Každý senzor umožňuje zaznamenávání těchto hodnot:

- časová značka záznamu,



- jízdní pruh, ve kterém se událost vyskytla,
- rychlost vozidla,
- délka vozidla.

Přesnost senzoru pro určení obsazenosti jízdního pruhu uváděná výrobcem je 98% – 99%, minimálně však 90%. Přesnost určení rychlosti vozidla je  $\pm 5\text{ km/h}$  (pro 90% měření).



Obrázek 9: Instalace snímače Wavetronix (zdroj [20])

### 3 Analýza a realizace monitorovací aplikace

Na základě zákonů uvedených v sekci 2.1 se tady rozhoduje, jakým způsobem monitorovat parkoviště a poskytovat tak dále informace o jejich obsazenosti řidičům. Úkolem tedy je zjistit, zda je potřeba instalace nových zařízení na všechny parkoviště, nebo postačí pouze využití dnes již zaběhnutého mýtného systému a informací z něj, pro monitorování pohybu vozidel a zjišťování obsazenosti.

Protože je však největším problémem parkování na parkovištích příslušejících dálnicím, bude se další zabírat především těmito parkovišti a parkoviště mimo dálnice se zatím nebudou uvažovat. Mimo dálnice nevzniká problém s dlouhými úseky, bez možnosti zastavení. Pro vlastní analýzu byla vybrána jedna z nejfrekventovanějších českých dálnic, a sice dálnice D1.

Bylo třeba provést klasifikaci parkovišť na této dálnici vůči mýtnému systému, tj. vůči mýtným branám a také zjistit jaká data tyto brány poskytují a jakým způsobem je možné je využít.

Mýtný systém pro monitorování se začal uvažovat později. Nejprve se uvažovalo o instalaci nových snímacích prvků na parkoviště, které by tyto informace poskytovaly. Tyto snímače byly již instalovány na dvě parkoviště na dálnici D5. Data z těchto snímačů měly být využity ve spojení s daty z mýtných bran, proto bylo zapotřebí zjistit, zda je možné tyto data mezi sebou nějakým způsobem párovat.

Všechny tyto data pak musejí být nějakým způsobem shromažďovány a poskytovány dalším subjektům pro další zpracování, případně poskytnuty přímo řidičům. Pro tento účel byla zpracována aplikace, která má zastřešovat všechny tyto funkce, tj. sběr a poskytování informací.

Z mýtného systému byly k dispozici historická data. To vedlo k vyvinutí simulátoru, který umožňuje simulaci času pro práci s těmito historickými daty. Simulátor má však také možnost simulovat virtuální vozidla a tím nahradit alespoň částečně potřebu dat pro zpracování.

## 4 Klasifikace parkovišť

Před samotným využíváním dat z mýtných bran bylo potřeba provést analýzu umístění parkovišť na dálnici a polohu mýtných bran relevantních k těmto parkovištím.

U parkovišť na dálnicích je možné evidovat několik základních parametrů:

- Kilometr – kilometr dálnice kde se nachází
- Benzínová pumpa – ne všechny parkoviště pro odpočinek jsou u benzínových pump
- Restaurace – zda je na parkovišti k dispozici nějaká forma občerstvení nebo stravování
- Ubytování – určuje možnost přespání
- Mýtná brána před a za – specifikuje, která brána se nachází na dálnici před resp. za parkovištěm (poloha je vztažena vůči směru jízdy)

Dále je potřeba rozlišovat i směr jízdy. Jednotlivá parkoviště nemusejí být vždy párová, tzn., že např. na dálnici ve směru jízdy z města A do města B je parkoviště, ale v opačném směru, tedy ve směru z města B do města A již parkoviště být nemusí.

Nejedná se ale pouze o existenci samotného parkoviště. Je možné, aby parkoviště bylo párové, ale v jednom směru bude na parkoviště ubytování a ve směru druhém už ubytování k dispozici nebude. To však neplatí jen pro ubytování, ale celkově pro všechny parametry, které jsou u parkovišť sledovány.

### 4.1 Parkoviště na dálnici D1

Ve dvou tabulkách 2 a 3 jsou uvedena všechna parkoviště na dálnici D1.

Z tabulky 2, která ukazuje parkoviště v jízdním směru z Hulína do Prahy, je patrné, že téměř všechna parkoviště (v tomto případě 21 z 26) jsou vystavěna okolo benzínových pump.

Pouze jedno parkoviště není vybaveno žádnou možností stravování, tzn., není zde k dispozici ani rychlé občerstvení ani restaurace. Jedná se o parkoviště na 138 kilometru s názvem Kochanov. Toto parkoviště je vyhrazeno pouze pro zastavení a odpočinek bez jakýchkoliv dalších služeb.

Ubytování na parkovištích je velice zřídka zařízeno jako motel nebo hotel. Řidiči dálkových nákladních vozidel mají většinou své vozidlo přímo v kabině vybavené lehátkem právě pro případ spánku v době odpočinku.

Pro srovnání, druhý směr z Prahy do Hulína 3, ukazuje rozdílnost v počtu a ve vybavenosti parkovišť v závislosti na směru. Byť se jedná o stejnou dálnici, je v jednom směru 26 a ve směru opačném jen 21 parkovišť.

Některá parkoviště navíc sdílejí své prostory mezi směry. Příkladem takového parkoviště je Střechov, kde vybavení parkoviště v obou směrech je totožné. To je dáno tím, že mezi jednotlivými směry je možné libovolně přejíždět tam i zpět spojovací silnicí, která

ID	km	Název	Benzínka	Restaurace	Ubytování	Brána před	Brána za
P201	4,5	Újezd u Průhonic	Ano	Ano	Ne	D01032	D01022
P202	5,0	Club Hotel Průhonice	Ne	Ano	Ano	D01032	D01022
P203	6,2	Průhonice	Ano	Ano	Ne	D01032	D01022
P204	10,0	Nupaky	Ano	Ano	Ne	D01042	D01032
P205	21,0	Kunice	Ne	Ano	Ne	D01062	D01052
P206	30,0	Hvězdonice	Ano	Ano	Ne	D01072	D01062
P207	41,0	U rybiček	Ne	Ano	Ne	D01092	D01082
P208	52,2	Střechov	Ano	Ano	Ano	D01102	D01092
P209	58,0	Kalná	Ano	Ano	Ne	D01112	D01102
P210	72,2	Dunice	Ano	Ano	Ne	D01132	D01122
P211	81,0	Speřice	Ne	Ano	Ne	D01142	D01132
P212	88,9	Humpolec	Ano	Ano	Ne	D01142	D01132
P213	96,0	Mikulášov	Ano	Ano	Ne	D01152	D01142
P214	111,0	Pávov	Ano	Ano	Ne	D01162	D01152
P215	121,8	Jamenský potok	Ne	Ano	Ne	D01182	D01172
P216	138,0	Kochanov	Ne	Ne	Ne	D01192	D01182
P217	144,7	Velké Meziříčí	Ano	Ano	Ne	D01202	D01192
P218	166,9	Devět Křížů	Ano	Ano	Ne	D01242	D01232
P219	185,0	Popůvky	Ano	Ano	Ne	D01262	D01252
P220	187,6	Troubsko	Ano	Ano	Ne	D01272	D01262
P221	193,0	Brno-Lískovec	Ano	Ano	Ne	D01282	D01272
P222	198,6	Brno-Tuřany	Ano	Ano	Ne	D01302	D01292
P223	206,5	Rohlenka	Ano	Ano	Ano	D01312	D01302
P224	227,0	Vyškov	Ano	Ano	Ne	D01352	D01342
P225	319,0	Vražné	Ano	Ano	Ne	D01502	D01494
P226	344,5	Klimkovice	Ano	Ano	Ne	D01542	D01532
P227	368,4	Antošovice	Ano	Ano	Ne	D01602	D01592

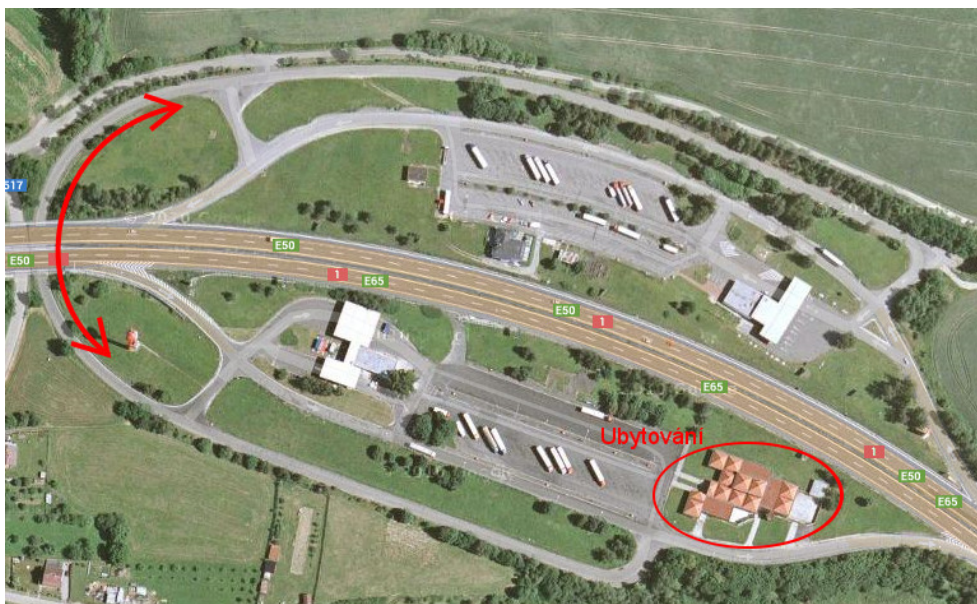
Tabulka 2: Parkoviště na D1 ve směru Hulín → Praha

ID	km	Název	Benzínka	Restaurace	Ubytování	Brána před	Brána za
P101	4,5	Újezdu Průhonice	Ano	Ano	Ne	D01021	D01031
P102	6,2	Průhonice	Ano	Ano	Ne	D01021	D01031
P103	10,0	Nupaky	Ano	Ano	Ne	D01031	D01041
P104	26,0	Poddubí	Ne	Ano	Ano	D01061	D01071
P105	33,0	Bělčice	Ano	Ano	Ne	D01061	D01071
P106	44,4	Blanice	Ano	Ano	Ne	D01091	D01101
P107	52,2	Střechov	Ano	Ano	Ano	D01091	D01101
P108	81,0	Speřice	Ne	Ano	Ne	D01131	D01141
P109	88,9	Humpolec	Ne	Ano	Ne	D01131	D01141
P110	96,0	Mikulášov	Ano	Ano	Ne	D01141	D01151
P111	111,0	Pávov	Ano	Ano	Ano	D01151	D01161
P112	121,8	Jamenský potok	Ano	Ano	Ano	D01171	D01181
P113	139,0	Kochanov	Ano	Ano	Ne	D01181	D01191
P114	144,7	Velké Meziříčí	Ano	Ano	Ne	D01191	D01201
P115	166,9	Devět Křížů	Ano	Ano	Ne	D01231	D01241
P116	187,6	Troubsko	Ano	Ano	Ne	D01261	D01271
P117	198,6	Brno-Tuřany	Ano	Ano	Ne	D01291	D01301
P118	206,5	Rohlenka	Ano	Ano	Ano	D01301	D01311
P119	318,0	Vražné	Ano	Ano	Ne	D01493	D01501
P120	344,5	Klimkovice	Ano	Ano	Ne	D01531	D01541
P121	368,4	Antošovice	Ano	Ano	Ne	D01591	D01601

Tabulka 3: Parkoviště na D1 ve směru Praha → Hulín



vede pod dálnicí. Na obou stranách dálnice jsou sice parkovací místa, ale ubytování je pouze na jedné straně (směr Praha - Hulín) a k tomuto ubytovacímu zařízení je možné se dostat právě po spojovací silnici, viz obrázek 10.



Obrázek 10: Spojovací silnice mezi parkovišti vrůžných směrech (zdroj: maps.google.com)

Další zajímavostí je umístění dvou a více parkovišť mezi dvěma mýtnými branami. Takovéto parkoviště mají stejnou příjezdovou a stejnou odjezdovou bránu. Na dálnici D1 je takovýchto parkovišť několik. Ve směru Hulín – Praha jsou to tato parkoviště:

- Újezd u Průhonic, Club Hotel Průhonice a Průhonice – všechny tato parkoviště mají stejnou příjezdovou bránu s označením D01032 a stejnou odjezdovou bránu s označením D01022
- Speřice a Humpolec – společná příjezdová brána je D01142 a odjezdová D01132

Ve směru Praha – Hulín jsou to tato parkoviště:

- Újezd u Průhonic a Průhonice – společná příjezdová resp. odjezdová jsou D01021 resp. D01031
- Poddubí a Bělčice – společné brány – D01061 a D01071
- Blanice a Střechov – společné brány – D01091 a D01101
- Speřice a Humpolec – společné brány - D01131 a D01141

## 4.2 Mýtné brány, exity a parkoviště

V případě rozlišování pozic mýtných bran vůči parkovištím je se rozlišují dva typy mýtných bran a sice každé parkoviště má k sobě příslušející „příjezdovou“ a „odjezdovou“ mýtnou bránu.

Pod pojmem „příjezdová“ mýtná brána se rozumí taková brána, která je ve směru jízdy před tímto parkovištěm a „odjezdová“ brána je ve směru jízdy za sledovaným parkovištěm.

Jako další velice důležitý parametr jsou exity. V tomto případě je potřeba sledovat polohu exitů nejenom vůči parkovištím, ale zároveň vůči mýtným branám. Mohou nastat tyto varianty umístění bran, exitů a parkovišť:

1. Příjezdová brána → exit → parkoviště → odjezdová brána → exit



Obrázek 11: Exit za příjezdovou bránou

2. Exit → příjezdová brána → parkoviště → exit → odjezdová brána



Obrázek 12: Exit před odjezdovou bránou

3. Příjezdová brána → exit → parkoviště → exit → odjezdová brána



Obrázek 13: Exity za příjezdovou a před odjezdovou bránou

4. Exit → příjezdová brána → parkoviště → odjezdová brána → exit



Obrázek 14: Exity před příjezdovou a za odjezdovou bránou

Na základě tohoto faktu jsou sledovány u parkovišť další dva parametry:

- Exit před – udává, zda je exit umístěn mezi parkovištěm a příjezdovou bránou
- Exit za – tento parametr naopak uvádí, zda se exit nachází mezi parkovištěm a odjezdovou bránou.

V případě sledovaných parkovišť na dálnici D1 jsou uvedeny polohy exitů v tabulce 4 a tabulce 5.

Podle analýzy parkovišť na sledované dálnici většina z nich splňuje rozložení exitů mezi bránami podle bodů 1 a 2, tj. že mezi každými dvěma exity je právě jedna brána. Najdou se však i takové parkoviště, které mají mezi sebou a bránou exit jak před sebou, tak i za sebou (bod 3). Jedná se o parkoviště Nupaky v obou směrech a dále o parkoviště Rohlenka ve směru Hulín – Rohlenka.

Rozložení parkovišť podle bodu 4 je opět pozorovatelná v obou směrech jízdy na dálnici D1. Ve směru Praha – Hulín se jedná o parkoviště Speřice a Humpolec a v opačném směru je parkoviště Dunice.

### 4.3 Sledovaná parkoviště se senzory

Mimo sledovanou dálnici D1, hlavní tepnu vedoucí napříč Českou republikou, byly vybrány dvě parkoviště u Plzně na dálnici D5 pro instalaci snímačů. Parametry obou parkovišť jsou uvedené v tabulce 6.

Snímače jsou umístěny na sloupech veřejného osvětlení 15 tak, aby snímaly příjezd na parkoviště a odjezd z parkoviště. Přesné umístění snímačů je patrné na obrázku, kde je každý z nich vyznačen červeným bodem.



Obrázek 15: Rozmístění a označení senzorů u parkovišť (zdroj: maps.google.com)

ID	Název	Exit před	Exit za
P101	Újezdu Průhonic	Ne	Ano
P102	Průhonice	Ne	Ano
P103	Nupaky	Ano	Ano
P104	Poddubí	Ne	Ano
P105	Bělčice	Ano	Ne
P106	Blanice	Ne	Ano
P107	Střechov	Ano	Ne
P108	Speřice	Ne	Ne
P109	Humpolec	Ne	Ne
P110	Mikulášov	Ano	Ne
P111	Pávov	Ano	Ne
P112	Jamenský potok	Ano	Ne
P113	Kochanov	Ano	Ne
P114	Velké Meziříčí	Ano	Ne
P115	Devět Křížů	Ne	Ano
P116	Troubsko	Ne	Ano
P117	Brno-Tuřany	Ne	Ano
P118	Rohlenka	Ano	Ne
P119	Vražné	Ne	Ano
P120	Klimkovice	Ne	Ano
P121	Antošovice	Ne	Ano

Tabulka 4: Exity mezi branami a parkovišti Praha → Hulín

ID	Název	Exit před	Exit za
P201	Újezdu Průhonic	Ano	Ne
P202	Club Hotel Průhonice	Ano	Ne
P203	Průhonice	Ne	Ano
P204	Nupaky	Ano	Ano
P205	Kunice	Ano	Ne
P206	Hvězdonice	Ne	Ano
P207	U rybiček	Ne	Ano
P208	Střechov	Ne	Ano
P209	Kalná	Ne	Ano
P210	Dunice	Ne	Ne
P211	Speřice	Ne	Ano
P212	Humpolec	Ne	Ano
P213	Mikulášov	Ne	Ano
P214	Pávov	Ne	Ano
P215	Jamenský potok	Ne	Ano
P216	Kochanov	Ne	Ano
P217	Velké Meziříčí	Ne	Ano
P218	Devět Křížů	Ano	Ne
P219	Popůvky	Ne	Ano
P220	Troubsko	Ano	Ne
P221	Brno-Lískovec	Ano	Ne
P222	Brno-Tuřany	Ano	Ne
P223	Rohlenka	Ano	Ano
P224	Vyškov	Ano	Ne
P225	Vražné	Ano	Ne
P226	Klimkovice	Ano	Ne
P227	Antošovice	Ano	Ne

Tabulka 5: Exity mezi branami a parkovišti Hulín → Praha

	Plzeň – Praha	Praha – Plzeň
ID	PP001	PP101
Kilometr	83,4	83.4
Benzínová pumpa	Ano	Ano
Restaurace	Ano	Ano
Ubytování	Ne	Ne
Brána před	D05152	D05141
Brána za	D05142	D05151
Exit před	Ano	Ne
Exit za	Ne	Ano
ID snímače na příjezdu	RP_IN   13   17163	PR_IN   11   4832
ID snímače na výjezdu	RP_OUT   14   9235	PR_OUT   12   17161

Tabulka 6: Parkoviště s instalovanými snímači Wavetronix

## 5 Obecná datová analýza vstupů

V této kapitole budou postupně rozebrány datové výstupy, které jsou k dispozici z mýtného systému. Jako další část pak bude následovat popis parametrů, které jsou jako výstupy ze senzorů umístěných na parkovištích a také vyhlazovací algoritmy, které mají za úkol řešit neexistující vozidla vzniklá vlivem rušení a odrazem signálu u těchto senzorů.

### 5.1 Datová analýza mýtných bran

Tato kapitola je blíže rozvedena v neveřejné části. Obsahuje informace o analýze dat, které jsou k dispozici z mýtných bran - výstupy, identifikace bran a jejich umístění.

### 5.2 Datová analýza snímačů na parkovištích

U vybraných záchytných parkovišť jsou instalovány snímače na příjezdu k parkovišti a na odjezdu z parkoviště. Každý z těchto senzorů poskytuje následující sadu informací pro každé projíždějící vozidlo:

- Sequence – jednoznačný identifikátor záznamu
- Date – čas a datum záznamu s přesností na milisekundy
- Duration – doba, jak dlouho bylo vozidlo senzorem snímáno, tj. doba od vjezdu vozidla do zorného pole snímače až po opuštění tohoto pole
- Speed – rychlost projíždějícího vozidla
- Length – naměřená délka projíždějícího vozidla
- LaneID – číslo pruhu, ve kterém bylo vozidlo zaznamenáno (senzory jsou instalovány u příjezdů a odjezdů, kde je monitorován pouze jeden pruh, proto je toto číslo vždy 0)

Ne u všech záznamů poskytnutých právě těmito senzory se jedná o skutečné vozidla. V některých případech mohou senzory zaznamenat tzv. duchy, tj. odraz signálu od vozovky po odrazu od vozidla apod. Pro účely odstranění neexistujících vozidel byly navrženy tři vyhlazovací algoritmy, které mají za úkol ze záznamů poskytnutých senzory vytvořit sadu dat, která by měla odpovídat skutečnému vozidlu, které před senzorem projelo.

Tyto algoritmy budou popsány v následujících podkapitolách.

#### 5.2.1 Detekce potencionálně neexistujících vozidel

(Algoritmus pro detekci potencionálně neexistujících vozidel pracuje vždy s jediným záznamem. Má za úkol vypořádat se s možnými odrazy měřícího signálu (mikrovln) od pevných částí infrastruktury v případě, kdy monitorované vozidlo nejede v přímé linii nebo dojde k zaznamenání odrazu z jiného důvodu.

Radar pracuje s minimální délkou přibližně 1,83m. To ve výsledku znamená, že jakýkoli záznam, a to jak odraz, tak i reálné vozidlo či jiný předmět, má minimální délku 1,83m. Na základě této délky se dále verifikuje, zda zaznamenaný prvek je věrohodný, nebo nikoli.

Věrohodnost se ověří na základě vyhodnocení těchto vstupních veličin, které radar poskytuje:

- Duration [ms] – doba trvání záznamu, tj. doba mezi tím, kdy radar zaznamenal, že v jeho zorném poli se nachází vozidlo a okamžikem, kdy vozidlo opustí zorné pole radaru.
- Speed [km/h] – rychlost vozidla zaznamenaná radarem při vjezdu vozidla do zorného pole.
- Length [m] – délka vozidla, kterou vrátí samotný radar (minimální délka je 1,83m).

V následující tabulce jsou uvedeny vstupní hodnoty naměřené radarem, na kterých bude popsáno fungování algoritmu:

Date	Duration [ms]	Speed [km/h]	Length [m]
2013-10-10 21:00:41.48	809	36,33203	6,3476562
2013-10-10 21:01:08.269	1201	17,710938	4,0859375
2013-10-10 21:02:12.016	2491	6,296875	2,53125
2013-10-10 21:02:12.789	328	6,296875	1,828125
2013-10-10 21:02:14.261	2925	4,65625	1,9570312
2013-10-10 21:02:14.926	185	4,65625	1,828125
2013-10-10 21:03:28.837	1170	20,042969	4,6914062
2013-10-10 21:03:46.28	1197	28,070312	7,5078125

Tabulka 7: Příklad hodnot pro určení existujících vozidel

Ze vstupních dat se vypočítají následující hodnoty:

- WT\_Speed [m/s] – převod původní hodnoty Speed z km/h na m/s.
- WT\_Duration [s] – převod Duration z milisekund na sekundy
- Length from duration [m] – délka vozidla, která je vypočtená z rychlosti WT\_Speed a doby snímání WT\_Duration podle vztahu  $WT\_Speed \times WT\_Duration$
- Difference [m] – rozdíl mezi délkou vypočtenou a délkou, kterou naměřil radar

Vypočtené hodnoty jsou uvedeny v tabulce:

Radar vždy naměří minimální délku přibližně 1,83m. Tato hodnota odpovídá dopočetnému zpoždění rychlosti signálu, který putuje mezi měřenou překážkou (v našem případě jedoucí vozidlo) a samotným radarem. Pokud tedy srovnáme rozdíl délky vypočtené a naměřené, obdržíme hodnotu vždy okolo 1,83m.



WT_speed [m/s]	WT_duration [s]	Length from duration [m]	Difference [m]
10,09303793	0,809	8,165267689	1,8176
4,920098576	1,201	5,90903839	1,8231
1,749271875	2,491	4,357436241	1,8262
1,749271875	0,328	0,573761175	-1,2544
1,29350625	2,925	3,783505781	1,8265
1,29350625	0,185	0,239298656	-1,5888
5,567936788	1,17	6,514486042	1,8231
7,797932674	1,197	9,33412541	1,8263

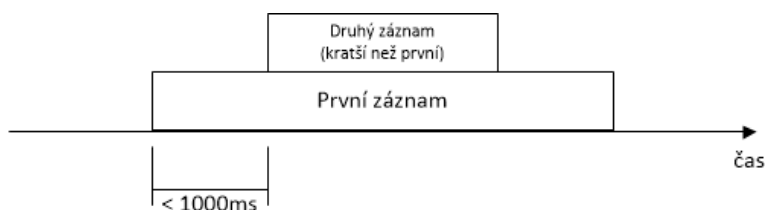
Tabulka 8: Přepočtené hodnoty pro určení existujících vozidel

V případě, kdy se jedná o chybný signál, je vypočtený rozdíl záporný. Dokonce i samotná vypočtená délka Length from duration je velice malá a neodpovídá délce žádného vozidla nebo motocyklu, které mají povolení k vjezdu na dálnici. Z tohoto faktu se tedy dá vyvodit závěr, že takto dopočtené záznamy je možné odstranit jako chybné.

V dříve uvedených tabulkách 7 a 8 jsou takovéto záznamy vyznačeny v odpovídajících si řádcích a tyto řádky nejsou pro pozdější zpracování brány v potaz.

### 5.2.2 Detekce překrývajících se vozidel

Algoritmus pro detekci překrývajících se vozidel pracuje se dvěma nebo více záznamy. V tomto případě se zpracovávají záznamy, které jsou v čase blízko sebe. Tento algoritmus má možnost volby časové vzdálenosti mezi záznamy – v našem případě se jedná o rozmezí 1 vteřina (1000ms). Hodnota je zvolena na základě předpokladu, že dvě vozidla mezi sebou musí dodržovat bezpečnou vzdálenost a nemohou tak projet v kratším intervalu než 1 vteřina z důvodu možné kolize. Situace je naznačena na obrázku 16.



Obrázek 16: Překrývající se vozidla

V této situaci se vyhledávají záznamy, které mají časovou prodlevu nižší než 1 vteřina a zároveň je jeden ze záznamů delší než záznam druhý, viz tabulka 9.

Jak je ze vzorku dat patrné, první dva záznamy mají časovou prodlevu pouze 96ms. Záznam obdrženy dříve má délku více než 16m, zatímco druhý, pozdější záznam, má délku necelých 12m. V reálu by to znamenalo, že obě tato vozidla jedou vedle sebe. Protože je však radar instalován na odbočovacím pruhu, kde je možný průjezd pouze jednoho

Date	Duration [ms]	Speed [km/h]	Length [m]
2013-10-11 06:48:52.679	2935	22,21875	16,289062
2013-10-11 06:48:52.775	2074	23,726562	11,84375
2013-10-11 06:49:58.555	1339	18,636719	5,109375
2013-10-11 06:50:28.411	954	18,636719	3,1171875
2013-10-11 06:50:54.12	2279	28,839844	16,4375
2013-10-11 06:53:19.949	1942	31,816406	15,339844

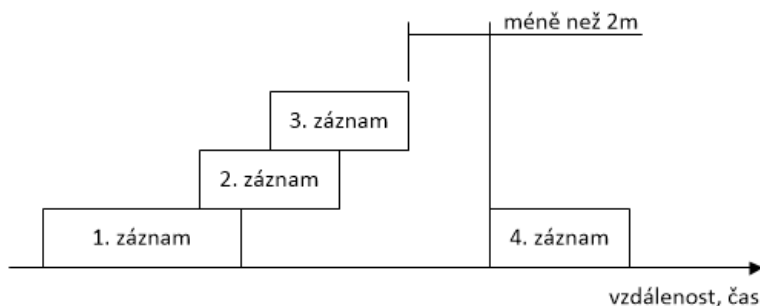
Tabulka 9: Příklad hodnot pro určení překrývajících se vozidel

vozidla zároveň, je tato situace nemožná. Na základě těchto úvah můžeme odstranit kratší záznam.

Odstranění kratšího záznamu volíme z důvodu, že k dané situaci nastává nejčastěji u nákladních automobilů, které mají kovový návěs. Signál se od takového návěsu v určitých případech odráží jiným způsobem a vytváří tak neexistující záznamy.

### 5.2.3 Algoritmus spojování záznamů

Tento algoritmus pracuje s dvěma a více záznamy a stejně jako u předchozího algoritmu se i tento zabývá záznamy, u kterých je časová prodleva nižší než 1 vteřina. Rozdíl je však v délce jednotlivých záznamů, kdy delší záznam plně nepřekrývá ostatní záznamy. Opět je situace znázorněna obrázkem 17.



Obrázek 17: Záznamy pro spojení

V tabulce je vzorek dat, na kterých bude vysvětleno fungování algoritmu.

Date	Duration [ms]	Speed [km/h]	Length [m]
2013-10-07 05:27:30.927	831	26,43	4,28
2013-10-07 05:27:31.503	561	26,43	2,29
2013-10-07 05:27:31.508	561	26,43	2,30
2013-10-07 05:27:32.083	867	26,43	4,54

Tabulka 10: Příklad hodnot pro spojení záznamů

Vstupní parametry v tomto případě jsou:

- Date – čas záznamu s přesností na milisekundy.
- Speed [km/h] – rychlost vozidla naměřená radarem.

Na základě vstupních parametrů jsou vypočteny tyto hodnoty

- Time between events – časový rozdíl mezi jednotlivými záznamy, bere se vždy čas dvou po sobě jdoucích záznamů a určí se jejich rozdíl.
- WT\_Speed [m/s] – původní parametr Speed je převeden z km/h na m/s.
- Distance between events – časů mezi událostmi (Time between events) a rychlosti (WT\_Speed) je určena vzdálenost mezi jednotlivými událostmi. Vzdálenost je uvedena v rozmezí od začátku jednoho záznamu do začátku druhého záznamu.

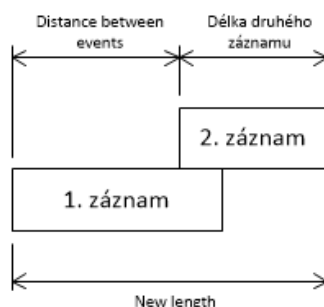
Tabulka 11 obsahuje vypočtené hodnoty (řádky korespondují s řádky v předchozí tabulce 10).

New length [m]	Time between events	WT_speed [m/s]	Distance between events
	0,58	7,34	4,19
6,48	0,6	7,34	4,41
6,7	1,15	7,34	8,44
12,98		7,34	

Tabulka 11: Příklad postupného spojení záznamů a výsledná délka

Nyní budeme předpokládat, že přijíždějící vozidla dodržují minimální vzdálenost 2m. Pokud je tedy vypočtené vzdálenost mezi záznamy větší než délka vozidla naměřeného radarem + minimální vzdálenost 2m ( $\text{Distance between events} > (\text{Length} + 2)$ ), jedná se o samostatné vozidlo a záznam je ponechán beze změny.

V opačném případě, tedy pokud je vzdálenost mezi záznamy menší než délka vozidla + 2m ( $\text{Distance between events} < (\text{Length} + 2)$ ), jsou dva záznamy jedno a totéž vozidlo a můžeme je tedy spojit do jednoho.



Obrázek 18: Spojení dvou záznamů do jediného

Spojení záznamů se provede tak, že se sečte vzdálenost mezi záznamy s délkou druhého záznamu, první záznam se odstraní a délka druhého záznamu (viz obrázek 18) je nahrazena novou délkou. Pro ilustraci je nová délka uvedena v tabulce 11 v prvním sloupci.

Takto se postupně projdou všechny sousední záznamy, které splňují výše uvedené podmínky, a buď se spojí, nebo ponechají beze změn. V tabulce je v posledním řádku uvedena celková délka vozidla dopočtená po spojení celkem čtyř záznamů do jediného.

## 6 Poskytované výstupy

Tato kapitola bude věnována výstupům, které bude možné po zpracování údajů z mýtných bran a senzorů, dále poskytovat řidičům, případně dalším subjektům. Bude zde také probrána možnost monitorování času stráveného na parkovišti a vytváření časových profilů pro mýtné brány a parkoviště.

### 6.1 Výstupy na základě mýtných bran

Na základě nařízení Evropské komise č. 885/2013, článek 6 [2] o šíření informací, musí poskytovatel zobrazovat informace o:

- přinejmenším dvou nejbližších bezpečných a chráněných parkovacích místech u koridoru v rámci přibližně 100 km,
- dostupnosti parkovacích míst v prioritní zóně alespoň u dvou nejbližších parkovacích ploch v rámci přibližně 100km.

Tuto podmínku je možné splnit díky přesně určené poloze všech parkovišť na dálnicích a zpoplatněných silnicích. Nutností však bude vybavení vozidla zařízením pro určení aktuální polohy (nejčastěji GPS navigace), které však bude také schopno zpracovávat data ze systému monitorování parkovišť. Toto zařízení pak systému poskytne aktuální polohu případně směr jízdy a systém po vyhodnocení těchto údajů vrátí do zařízení informace o všech dostupných parkovištích ve vzdálenosti maximálně 100 km od aktuální polohy.

Pro sledování obsazenosti parkoviště je potřeba znát dvě čísla – počet vozidel, které na parkoviště vjedou a počet vozidel, které z parkoviště odjedou. Rozdíl těchto dvou čísel dá pak v určitém časovém okamžiku počet vozidel, které se na parkoviště nacházejí.

Pokud se jedná o mýtné brány, je možné počet vozidel monitorovat přímo na nich. Příjezdová mýtná brána slouží jako počítadlo vozidel, které do daného úseku silnice, kde se také nachází parkoviště, vjedou. Odjezdová brána zase slouží jako počítadlo vozidel, které z daného úseku vyjedou.

Problém nastává v případě, kdy je mezi parkovištěm a některou z mýtných bran exit. Pokud nastane situace, kdy je exit mezi příjezdovou bránou a parkovištěm, může dojít k následujícím třem situacím:

1. vozidlo bude pokračovat od příjezdové brány k parkovišti, a buď na něm zastaví, nebo parkoviště mine a projede odjezdovou bránou,
2. vozidlo sjede na exitu hned za bránou a vůbec tedy nebude projíždět bránou odjezdovou,
3. vozidlo neprojede příjezdovou bránou, ale najede na dálnici až na nájezdu příslušející danému exitu a buď bude pokračovat dál na odjezdovou bránu, nebo zastaví na parkoviště a až pak bude pokračovat dále v cestě.

Pokud tedy bude monitorování parkoviště využívat jeden seznam, do které bude příjezdová mýtná brána vozidla přidávat a odjezdová brána bude naopak vozidla odebírat, bude toto měření bezchybné pouze v prvním případě.

První případ je ideální stav, kdy je možné bez problému určit stav obsazenosti parkoviště na základě průjezdů na příjezdové a odjezdové bráně. Nedochozí totiž k „únikům“ vozidel a každé vozidlo, které se objeví na příjezdové bráně, se také musí objevit na bráně odjezdové.

Ve druhém případě se vozidlo do seznamu přidá díky příjezdové bráně, ale už v tomto seznamu zůstane, protože jej odjezdová brána nikdy nezaznamená a nemůže jej tak ze seznamu odstranit. Je však jisté, že toto vozidlo vůbec na parkovišti nebylo zaparkováno, jinak by totiž muselo projet odjezdovou bránou.

V případě třetím, dojde k situaci, kdy bude odjezdová brána odstraňovat ze seznamu vozidla, která v něm nejsou, ale u takovýchto vozidel není jisté, zda projížděli parkovištěm nebo jej jenom minula.

Další problémovou situací je rozmístění, kdy je exit mezi odjezdovou bránou a parkovištěm. Tato situace může mít tyto tři průběhy:

1. vozidlo projede příjezdovou bránou, bude pokračovat mimo parkoviště nebo na něm na chvíli zastaví a poté projede bránou odjezdovou,
2. vozidlo na parkoviště na chvíli zastaví nebo jej mine a pak opustí silnici na exitu a neprojede tak odjezdovou bránou,
3. vozidlo vůbec parkoviště nemělo ani na něm nezastavovalo, najelo na dálnici až na nájezdu příslušející danému exitu a projelo tak až odjezdovou bránou.

Opět se bude uvažovat použití seznamu pro obsazenost parkoviště, kdy jej příjezdová brána bude postupně plnit a odjezdová brána naopak postupně vyprazdňovat. První případ je opět ideální kdy není potřeba řešit zbylé vozidla v seznamu nebo naopak mazání neexistujících vozidel.

U druhého případu však není podchyceno, zda vozidlo vůbec parkovištěm projelo a není tak možné s tímto vozidlem správně kalkulovat.

Třetí případ je obdobou třetího případu z předchozího rozmístění. Opět by odjezdová brána odstraňovala ze seznamu vozidlo, které v něm vůbec není.

Další extrémní situací je možnost, kdy jsou exity umístěny mezi parkovištěm a oběma branami. V takovémto případě je možné spojit všechny body z předcházejících dvou rozložení a monitorování vozidel je tak značně zkreslené a neposkytuje věrohodné výsledky.

Ideálním stavem je rozložení, kdy na daném úseku mezi dvěma branami není žádný exit. V tomto případě nemají vozidla možnost přijet nebo odjet jinudy než právě přes brány příslušející danému parkovišti. Tuto situaci je možné znázornit prvním bodem z kteréhokoli předchozího rozložení a tedy, že nedojde k mazání neexistujících vozidel nebo v seznamu nebudou zůstat vozidla, které na daném úseku již dávno nejsou.

## 6.2 Výstupy ze senzorů

U senzorů umístěných na parkovišti je také možnost zavést seznam vozidel. V takovém případě je však třeba podotknout, že vozidla již nemají možnost tento úsek opustit jinudy než výjezdem z parkoviště, kde je však umístěn další senzor. Výjimkou můžou být parkoviště, ke kterým vedou postranní příjezdové cesty. Ani jedno ze sledovaných parkovišť však takovou cestu nemá.

Parkoviště vybavené senzory tak kopíruje první ideální stav od všech rozmístění mýtných bran, exitů a parkovišť a je tedy možné nahradit seznam pouze čítačem, kdy příjezdový senzor bude čítač zvyšovat a objezdový senzor jej bude snižovat.

Dále senzory zaznamenávají všechny vozidla, tj. i vozidla, které nepodléhají povinnosti pravidelných odpočinků a tato vozidla je možné ze sledování vyloučit. Budou se tedy započítávat pouze vozidla o délce minimálně 9 m (kratší nákladní vozidla s hmotností vyšší než 7,5 tuny se běžně nevyskytují) a ostatní se nebudou brát v potaz pro další zpracování.

## 6.3 Doba strávená v úseku mezi branami

Vozidla je možná identifikovat napříč branami. Tato skutečnost je blíže rozebrána v nevěřejné části a zde je pouze rozveden způsob využití této identifikace.

Pokud bude uvažován ideální stav, tedy vozidlo bude projíždět pod oběma branami příslušejícími parkovišti, je možné strávený čas rozdělit do čtyřech kategorií v závislosti na vybavení parkoviště:

- vozidlo od příjezdové brány přímo a přes parkoviště neprojíždí,
- vozidlo zastavuje na parkoviště pouze pro potřeby natankování, přičemž je možné na natankování uvažovat čas přibližně 30 minut (pouze v případě, že je k dispozici benzínová pumpa),
- vozidlo na parkovišti zastaví za účelem přestávky po 4,5 hodinách řízení na 45 minut,
- vozidlo na parkovišti zastaví na běžnou denní dobu odpočinku, tj. na 9 – 11 hodin.

Časy uvedené v předchozích čtyřech bodech jsou čisté časy, tzn., není v nich započítán čas potřebný na projetí vzdálenosti mezi oběma branami. Tento čas je určen podle vzorce  $\text{čas} = \text{vzdálenost} / \text{rychlost}$ , kde vzdálenost je absolutní hodnota rozdílu kilometrů umístění mýtných bran a rychlost je u dálnic 75 km/h, což je o 5 km/h nižší rychlost než je nejnižší povolená rychlost na dálnicích.

Možnost monitorovat čas průjezdů dává k dispozici také možnost částečně vyřešit problém s exity v úseku mezi branami.

V případě, kdy je exit mezi příjezdovou bránou a parkovištěm, dostane se vozidlo na seznam příjezdových vozidel, ale tím, že silnici na exitu hned opustí, by došlo k situaci, že by se již neobjevil na výjezdu a blokoval by tak jedno parkovací místo na parkovišti, i když by na něm ve skutečnosti nebyl. Zde se nabízí poměrně jednoduché řešení situace.



Pokud je vozidlo v seznamu delší dobu než 11 hodin, mělo by každou chvíli parkoviště opustit. Zaměstnavatelé totiž tlačí, aby byly přestávky co nejkratší a aby tak náklad dorazil na místo co nejdříve. Jakmile je tedy vozidlo v seznamu již 15 hodin a více, je možné ho ze seznamu vyřadit a označit, že vůbec na parkovišti nebylo.

U stejného rozložení exitu, brány a parkoviště, ale pro případ, že vozidlo najede na silnici nájezdem příslušejícím k danému exitu, není možné monitorovat dobu strávenou na úseku. Je však méně pravděpodobné, že krátce po najetí na placenou silnici by řidič zastavil na prvním parkovišti na dobu delší než 30 minut. Takže takováto vozidla nebude potřeba uvažovat.

Rozložení, kdy je exit mezi odjezdovou bránou a parkovištěm je řešeno obdobným způsobem. Jakmile je vozidlo zavedeno do seznamu, odpočítává se opět časový limit 15 hodin, po kterém se z tohoto seznamu odstraní. Opět je málo pravděpodobné, že by řidič zastavil na delší dobu na parkovišti a pak by toto hned dalším exitem opustil zpoplatněnou silnici.

Vozidla, která na tento úsek vjedou nájezdem příslušejícím k danému exitu, není potřeba započítávat do seznamu, nemají totiž žádnou možnost na parkovišti zastavit. Ostatní vozidla jsou kalkulována stejně jako v předchozím případě.

#### 6.4 Doba strávená v úseku mezi senzory

U mýtných bran je možná jednoznačná identifikace vozidla napříč branami a tím je možné měřit dobu, jakou toto vozidlo mezi dvěma branami strávilo. U senzorů umístěných na parkovištích tato možnost již není. Chybí zde totiž jakákoli možnost párování vozidla na příjezdovém a odjezdovém senzoru. Jediná možnost jak by bylo možné vozidla párovat je na základě jejich délky. Avšak s ohledem na nepřesnost měření délky vozidla, kdy při příjezdu vozidlo přibrzdí a jeho délka se tak může prodlužovat a při odjezdu, kdy vozidlo začíná akcelarovat a jeho délka se tak může zkrátit, není možné nikdy zcela přesně určit, zda se jedná o jedno a totéž vozidlo. Pokud by bylo měření na základě délky zcela přesné, je zde stále okolnost, kdy mohou na parkoviště přijet dvě vozidla stejného typu a délky a to by opět způsobilo zmatek při párování.

Senzory tak nemohou podávat informaci o době strávené na daném parkovišti. Na základě počtu vozidel, které přijedou a odjedou, je možné pouze určit počet vozidel nacházejících se na parkovišti v daný okamžik.

#### 6.5 Párování vozidel mezi branami a senzory

Další možnost je využití dat z mýtných bran a senzorů v jednom okamžiku, tj. pokusit se párovat vozidla projíždějící příjezdovou mýtnou bránou a příjezdovým senzorem. U této strategie nastává stejný problém jako u párování vozidel mezi dvěma senzory, tj. absence jednoznačného identifikátoru na senzorech není možné párovat vozidla z bran s vozidly na senzorech.

Párování podle délky je opět díky rozdílné rychlosti téměř nemožné. Vozidlo při průjezdu bránou má plnou rychlost, zatímco na senzoru již brzdí, rychlost je podstatně nižší a dochází ke zkreslení délek.

## 6.6 Profily mýtných bran

Na základě průjezdů vozidel branami je tedy možné pro brány vytvořit profily průjezdů. Je však zbytečné vytvářet profil počtu projetých vozidel např. pro celý jeden den, takový profil by byl nepraktický. Je tedy potřeba rozdělit den do kratších časových úseků. Zvolená délka časového úseku je 10 minut, tj. jeden den se rozdělí na 144 menších částí (24 hodin = 1440 minut) a pro každý desetiminutový interval se určí počet vozidel, které pod bránou projedou.

Neveřejná část práce ukazuje příklad z reálných dat, na kterém je ukázáno proč jsou profily zvoleny tímto způsobem.

Průměrný počet vozidel je však lepší ukládat jako interval s maximální a minimální hodnotou, tedy např. že v pondělí jsou průjezdy danou bránou v době 9:30 – 9:40 minimálně 10 a maximálně 15 vozidel. Tento způsob uložení dává pak větší svobodu pro určení tzv. katastrofických scénářů, kdy je možné počítat s nejvyšším průjezdem v danou dobu.

## 6.7 Profily parkovišť

Díky možnosti monitorovat časy strávené v úseku silnice z dat získaných z mýtných bran, je možné vytvořit také profily pro parkoviště.

Zde však není nutné se omezit pouze na vytvoření profilu s počtem vozidel, které parkovištěm projely, ale je možné tento počet ještě dále rozdělit na určitých kategoriích podle času stráveném na úseku silnice (časy jsou uvedené po odečtení doby potřebné pro ujetí vzdálenosti mezi příslušnými branami):

- 0 – 30 minut – tankování
- 30 – 90 minut – krátký odpočinek a stravování
- 90 – X minut – denní doba odpočinku (X je maximálně 15 hodin, pak je vozidlo vyhodnoceno, že vůbec na parkoviště nevjelo a silnici opustilo na výjezdu před odjezdovou mýtnou bránou)
- Celkový počet vozidel aktuálně se nacházejících na území parkoviště (součet všech vozidel, které na parkovišti v danou dobu stály)

Pomocí těchto kategorií je možné dále předpovědět, zda na parkovišti vozidla pouze zastaví, aby natankovala, nebo zastavují na delší dobu a je tak menší pravděpodobnost, že bude na parkovišti volné parkovací místo. Stejně jako u profilů mýtných bran je i profilů parkovišť třeba vyřešit způsob uložení a počet profilů. Proto i zde jsou vytvořeny profily pro každý den v týdnu zvlášť a každý den je dále rozdělený na intervaly po 10-ti minutách. V každém takovémto 10-ti minutovém intervalu jsou opět ukládány dvě hodnoty – minimální a maximální počet vozidel, které splňují dané podmínky.

## 6.8 Profily parkovacích senzorů

Data ze senzorů poskytují pouze neidentifikované vozidla. Je tedy možné sledovat pouze počet vozidel, které před daným senzorem projedou a profily senzorů tak budou odpovídat profilům mýtných bran co do rozdělení dne na časové intervaly a ukládání minimálních a maximálních hodnot průjezdů pro daný interval.

## 6.9 Predikce obsazenosti parkoviště

Informace pro řidiče o volných parkovacích místech se poskytují pro parkoviště do vzdálenosti 100km od aktuální polohy vozidla. Než však vozidlo k parkovišti dojede, může již být toto volné místo obsazené. Proto je potřeba provádět alespoň krátkodobou predikci obsazenosti parkoviště.

Aby bylo možné předvídat, zda bude parkoviště plně obsazené či nikoli, je zapotřebí znát několik faktorů:

- Kapacita parkoviště
- Aktuální počet vozidel na parkovišti
- Doba přibližného příjezdu na parkoviště
- Průměrný počet vozidel, které na parkoviště v danou dobu zaparkují.

V případě, kdy mezi branami a parkoviště není žádný exit, jsou všechny body pro predikci splněny a je tedy možné předpovídat obsazenost parkoviště. To je možné provést následujícím způsobem:

1. Od kapacity parkoviště se odečte aktuální počet vozidel, které na parkovišti jsou již zaparkované,
2. Z profilu pro parkoviště se vybere počet vozidel, které zaparkují na delší dobu (více než 30 minut) v době od zadání požadavku až do doby plánovaného příjezdu
3. Hodnoty z bodu 2 se sečtou a odečtou se od hodnoty vypočtené v bodě 1 a výsledek je předpokládaný počet volných parkovacích míst v době příjezdu na parkoviště.

Pro upřesnění predikce by bylo možné zavést ukládání počtu vozidel, které projedou příjezdovou branou za posledních X hodin. Tyto hodnoty se následně porovnají s odpovídajícími 10-ti minutovými intervaly a určí průměrná pozice v těchto intervalech, např. přes bránu podle profilu v jednom 10-ti minutové úseku projede minimálně 10 a maximálně 15 vozidel a aktuálně bylo pro tento časový úsek u brány zaznamenáno 12 vozidel, znamená to tedy, že hodnota 12 je 40% intervalu mezi 10 a 15 (50% by bylo 12,5 vozidla). Takto se vypočítá zpětně průměrná pozice v intervalu za poslední hodinu a tato hodnota se použije pro určení hodnot z bodu 2. Další postup je již stejný.

Pokud se bude brát v potaz parkoviště, které má mezi některou z příslušných bran exit, je problém s obdržením dat pro první bod – aktuální počet vozidel na parkovišti.

Není totiž možné kalkulovat s vozidly, které projedou příjezdovou bránou a jsou v úseku např. 1 hodinu čistého času, jako vozidla, která jsou na parkoviště zaparkovaná. Může totiž nastat situace, kdy některá z vozidel na parkovišti skutečně zastavila, ale některá z nich opustila silnici právě na příslušném exitu. To by znamenalo zkreslené informace pro další výpočet.

V tomto případě je možné určovat obsazenost parkoviště pouze na základě historických dat. Postup pro určení počtu vozidel na parkovišti může vypadat takto:

1. Z profilů pro parkoviště se vyberou za posledních 10 hodin (vozidlo může parkovat i 11 a více hodin) průměrné hodnoty parkujících vozidel na dobu delší než 90 minut a ty se sečtou
2. Z profilů se vyberou za posledních 30 minut průměrné hodnoty parkujících vozidel na dobu mezi 30 a 90 minutami a sečtou se
3. Hodnoty z bodů 1 a 2 se sečtou a odečtou se od kapacity parkoviště.

Pro přesnější určení obsazenosti je možné využít opět průměrnou hodnotu na intervalu jako v předchozím případě. Ovšem tato predikce je stále pouze přibližná.

Predikci na základě dat ze senzorů je možné provést obdobně jako predikci u mýtných bran, kdy v úseku nejsou žádná parkoviště. Chybí zde však data o době, jakou vozidla stráví na parkovišti. Výpočet tedy může vypadat následovně:

1. Čítač počtu vozidel na parkovišti se odečte od kapacity parkoviště, tím se získá aktuální volná plocha parkoviště
2. Z profilů pro příjezdový senzor se načtou průměrné průjezdy v době od aktuálního času do doby předpokládaného příjezdu a sečtou se
3. Z profilů pro odjezdový senzor se načtou průměrné průjezdy v době od aktuálního času do doby předpokládaného příjezdu a sečtou se
4. Výsledek z bodu 3 se odečte od výsledku z bodu 2 a výsledná hodnota se odečte od výsledku z bodu 1, tímto je určeno, kolik volných míst by na parkovišti mělo v danou dobu být.

Stejně jako v předchozích dvou případech se dá využít průměrných hodnot na intervalu pro přesnější získání dat z profilů v bodě 2 a 3.

## 6.10 Shrnutí možných poskytovaných výstupů

Na základě analýzy dat je tedy možné poskytnout koncovým uživatelům sadu těchto dat:

- Přesnou polohu parkoviště
- Vybavení parkoviště

- Benzínová pumpa: ANO/NE
  - Stravování: ANO/NE
  - Ubytování: ANO/NE
- Kapacita parkoviště
- Průměrnou historickou obsazenost parkoviště na základě profilů
- Pro parkoviště bez exitů mezi branami aktuální obsazenost parkoviště s krátkodobou predikcí
- Pro parkoviště s exity mezi branami pouze odhadovanou obsazenost parkovišť
- Pro parkoviště se senzory aktuální obsazenost parkoviště

## 7 Implementace

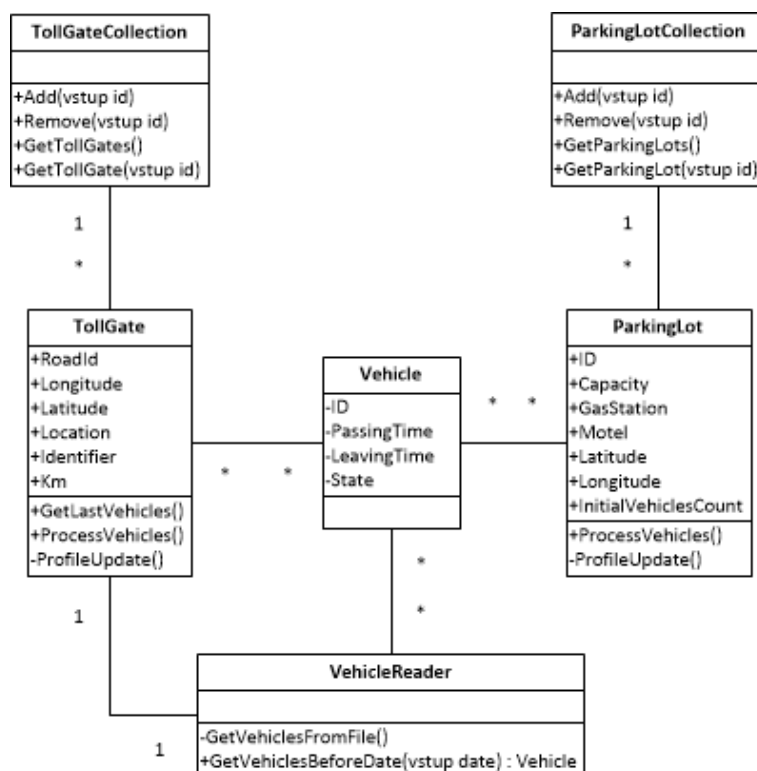
V implementační části se bude práce zabývat vytvořením aplikace pro správu dat o parkovištích, která je také schopna tyto data skrze jednotné rozhraní poskytovat dále. Také bude rozebrána aplikace pro simulování času a vytváření fiktivních vozidel pro testovací účely, její funkcionality a popis algoritmů v ní naimplementovaných.

### 7.1 Parking Manager

Parking manager je aplikace, který umožňuje spravovat parkoviště a jejich přidružené mýtné brány. Pomocí tohoto desktopového klienta je možné přidávat/odebírat mýtné brány do/ze systému.

Dále je v tomto klientovi také možno přidávat a odebírat parkoviště. Musí však být splněna podmínka, že jsou v systému zavedeny alespoň dvě mýtné brány. Bez mýtných bran není možné parkoviště vytvořit.

V klientské aplikaci jsou data o parkovištích i o branách načteny z databáze do dvou kolekcí, které mají za úkol udržovat konzistenci dat, tj. že v systému nebudou žádné dvě parkoviště nebo brány se stejným identifikátorem.



Obrázek 19: Třídní diagram pro parkoviště a mýtné brány

Na obrázku 19 je vidět třídní diagram, kde jsou znázorněny třídy pro mýtnou bránu, parkoviště a dvě jim příslušející kolekce. Dále jsou zde vidět další dvě třídy, jedna má za úkol držet informace o vozidle a druhá se stará o načítání projíždějících vozidel. Bližší specifikace jednotlivých tříd bude popsána dále.

### 7.1.1 Vehicle

Jednoduchá třída, která je pouze nositelem informace o vozidlu, které se v systému objevilo. U každého takového vozidla jsou vedeny tyto informace:

- ID – jednoznačný identifikátor vozidla, díky kterého je vozidlo identifikováno napříč celého systému,
- PassingTime – čas, kdy vozidlo projelo mýtnou bránu,
- LeavingTime – čas, kdy vozidlo projelo odjezdovou mýtnou bránu nebo odhadovaný čas jeho odjezdu pokud odjezdovou bránu neprojelo delší dobu než 15 hodin od projetí příjezdovou bránu,
- State – status vozidla, který může nabývat následujících hodnot:
  - NoPass – vozidlo neprojelo parkovištěm, ale rovnou pokračovalo k odjezdové bráně
  - Refuel – vozidlo zastavilo na parkoviště na dobu 0-30 minut
  - EatRest – vozidlo zastavilo na parkoviště na dobu 30-90 minut
  - Parked – vozidlo zastavilo na parkoviště na více než 90 a méně než 15 hodin

### 7.1.2 VehicleReader

Úkolem VehicleReader je pouze načíst data o vozidlech z CSV souboru pro danou bránu, vytvořit z těchto dat objekty Vehicle a seřadit je podle času. O to se stará metoda GetVehiclesFromFile.

Metoda GetVehiclesBeforeDate pak vrací seznam takto seřazených vozidel, ale pouze do času zadaného parametrem. Zbylé vozidla si ponechá v mezipaměti, a buď k nim přidá vozidla načtená při čtení dalšího CSV, nebo dále pracuje pouze s vozidly, které v tomto seznamu zbyla.

### 7.1.3 TollGate

Tato třída udržuje následující informace o mýtné bráně:

- Identifier – ID2 mýtné brány z tabulky zpoplatněných úseků (např. D01011),
- Location – ID1 ze stejného dokumentu jako ID2,
- Km – kilometr dálnice kde se brána nachází,



- RoadID – silnice, na které je brána umístěná (např. D1),
- Latitude a Longitude – GPS poloha brány.

Každá brána má pak za úkol držet si seznam vozidel, které přes ni projedou. Jakmile je zavolána metoda `GetLastVehicles`, brána vrátí jako výstup této metody seznam takovýchto vozidel, které jsou v bráně uloženy od posledního zavolání této metody.

Po každém zavolání metody `GetLastVehicles` dojde k vymazání seznamu posledních vozidel z brány. Mazání posledních vozidel je zde z důvodu snížení nároků na operační paměť, kdy by po delší době každá brána udržovala v paměti informace o velkém množství vozidel, se kterými se již dále pracovat nebude.

Metoda `ProcessVehicles` se stará o plnění seznamu vozidel, které bránou projela. Pro tento účel využívá na sebe navázaného objektu `VehicleReader` a jeho metodu `GetVehiclesBeforeDate`. Jakmile je seznam vozidel aktuální, zjistí se, zda má dojít k aktualizaci profilu a pokud ano, tak se profil aktualizuje (bude popsáno v kapitole Aktualizace profilů 7.1.7).

#### 7.1.4 ParkingLot

`ParkingLot` slouží jako držitel informací o parkovišti:

- ID – identifikátor parkoviště,
- `InTollGateID` – ID brány, která je uvažovaná jako příjezdová brána,
- `OutTollGateID` – ID brány, která je uvažovaná jako odjezdová brána,
- Capacity – kapacita parkoviště,
- GasStation – zda je na parkoviště k dispozici benzínová pumpa,
- Motel – přítomnost některé formy ubytování,
- Latitude a Longitude – GPS poloha parkoviště,
- `InitialVehiclesCount` – počet vozidel, které jsou na parkoviště v době spuštění systému. Systém se totiž může spustit v době, kdy na parkovišti již budou zaparkovaná nějaká vozidla a došlo by pak ke zkreslení informací.

Metoda `ProcessVehicles` této třídy je asi nejdůležitější metoda celého systému. Úkolem této metody je zpracování vozidel ze dvou mýtných bran (příjezdové a odjezdové) a určení, zda vozidla na parkovišti byla či nikoli a pokud přes parkoviště projížděla, tak na jak dlouho se na parkovišti zdržela.

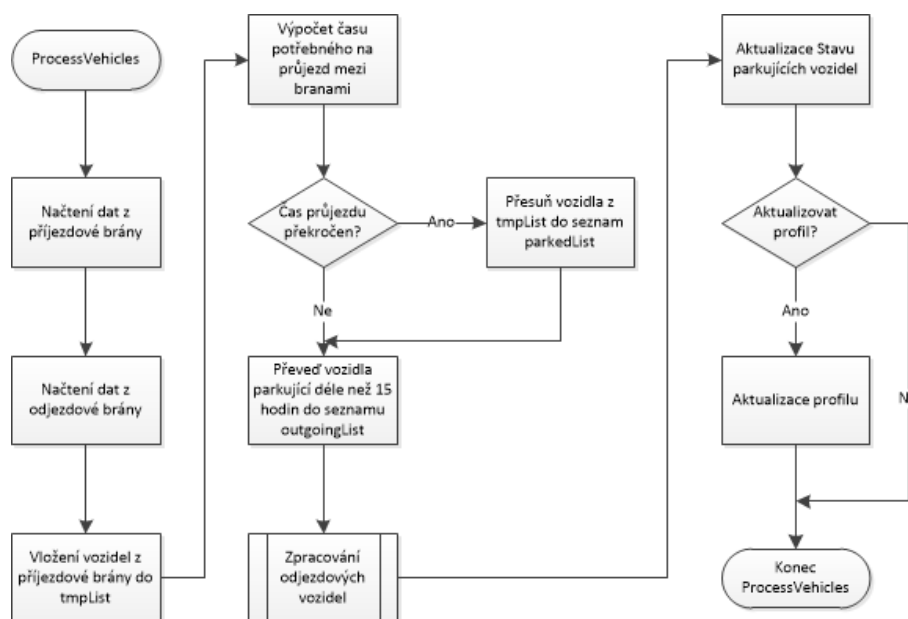
Celý proces je znázorněn dvěma vývojovými diagramy 20 a 21 a je možné jej shrnout tímto způsobem:

1. Načtou se vozidla z příjezdové a odjezdové brány.

2. Vozidla z příjezdové brány se vloží do seznamu vozidel, která jsou zatím na úseku mezi oběma branami (tmpList).
3. Ze vzdálenosti obou bran se vypočte čas potřebný na to, aby vozidlo urazilo vzdálenost od jedné brány ke druhé. Pro tento výpočet se využívá hodnota *km* obou bran a na překonání této vzdálenosti se uvažuje rychlost 75km/h. Tato rychlost je zvolena o 5km/h nižší než je nejnižší povolená rychlost na dálnici (při nižší rychlosti by mohlo být vozidlo vykázáno z dálnice pro zdržování provozu a porušení pravidel silničního provozu).
4. Projdou se všechny vozidla ze seznamu tmpList a pokud je aktuální čas větší, než součet času průjezdu příjezdovou bránou a času potřebného na ujetí vzdálenosti mezi branami, je vozidlo přesunuto z tmpListu do seznamu vozidel, které jsou na parkovišti (parkedList). U takového vozidla se navíc změní stav z NoPass na Refuel, čímž se zaznamená, že vozidlo projelo parkovištěm.
5. Všechna vozidla, která jsou v seznamu parkedList se projdou a zjistí se, zda rozdíl času příjezdu a aktuálního času není větší než 15 hodin (hodnota určená na základě několika pokusů). Pokud ano, je status těchto vozidel nastaven na NoPass a jsou zařazeny do seznamu vozidel z odjezdové brány (outgoingList) s časem vypočteným v bodě 3.
6. Proveďte se zpracování vozidel ze seznamu outgoingList.
7. Proveďte se aktualizace stavu u všech vozidel, která zbyla v seznamu parkedList a případně se zvýší jejich status z Refuel na EatRest nebo z EatRest na Parked.
8. Podle času poslední aktualizace profilu a nejdéle parkujícího vozidla se rozhodne, zda se bude profil parkoviště aktualizovat nebo ne (bude popsáno v kapitole Aktualizace profilů 7.1.7).
9. Konec metody ProcessVehicles.

Zpracování vozidel ze seznamu outgoingList je možné dále specifikovat takto:

1. Pro každé vozidlo ze seznamu outgoingList se zjistí, zda je v seznamu tmpList. Pokud ano, je z tohoto seznamu odstraněno a přejde se na zpracování dalšího vozidla.
2. Pokud vozidlo není v seznamu tmpList ani v parkedList, sníží se počítadlo inicializačního počtu vozidel. A přechází se ke zpracování dalšího vozidla.
3. Pokud je vozidlo v seznamu parkedList, je aktualizován LeavingTime, který je nastavený na PassingTime z outgoingList seznamu (čas, v kolik vozidlo projelo odjezdovou bránou).
4. Na základě rozdílu PassingTime a LeavingTime je určen skutečný stav vozidla (Refuel, EatRest nebo Parked).



Obrázek 20: Metoda ProcessVehicles

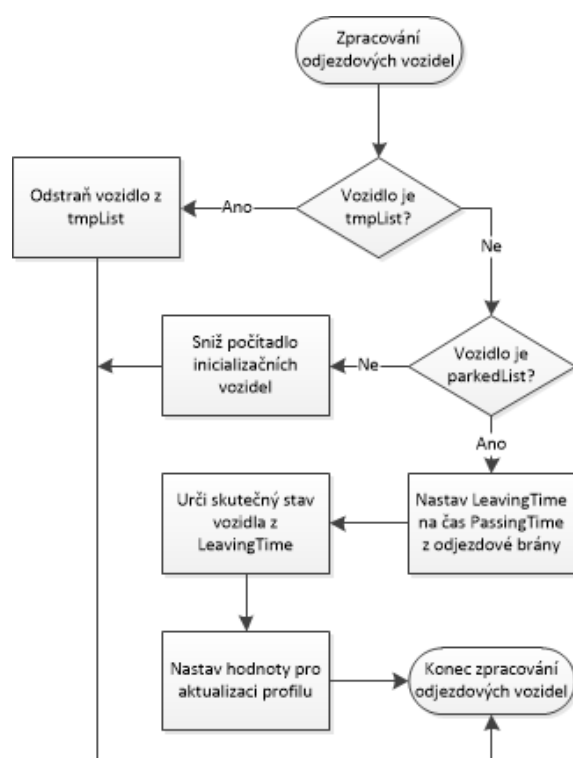
5. Podle stavu z bodu 4 je inkrementováno počítadlo pro aktualizaci určité části profilu parkoviště (bude popsáno v kapitole Aktualizace profilů 7.1.7).
6. Konec zpracování vozidel z outgoingListu.

### 7.1.5 TollGateCollection a ParkingLotCollection

Obě tyto kolekce pouze udržují seznam mýtných bran resp. parkovišť tak, aby nebyly v systému dvě mýtné brány resp. parkoviště se stejným identifikátorem.

Metody kolekcí a jejich funkcionalita:

- Add – umožňuje přidání další brány /parkoviště do kolekce, ale pouze tehdy, není-li již brána/parkoviště se stejným identifikátorem v kolekci již přítomna/no.
- Remove – odstraňuje bránu/parkoviště z kolekce na základě zadaného identifikátoru.
- GetTollGates/GetParkingLots – vrací seznam všech bran/parkovišť v kolekci.
- GetTollGate/GetParkingLot – vrátí jednu specifickou bránu/parkoviště na základě daného identifikátoru. Pokud neexistuje žádná brána/parkoviště se zadaným identifikátorem, nevrátí nic.



Obrázek 21: Zpracování odjezdových vozidel v metodě `ProcessVehicles`

### 7.1.6 Profily bran a parkovišť

Pro účely určení obsazenosti parkovišť je potřeba také ukládat data o počtu vozidel, která parkovištěm projela. Stejně tak je potřeba ukládat i data o projíždějících vozidlech u příslušných mýtných bran. K tomu slouží profily pro mýtné brány a parkoviště.

Profil je uložení počtu dat pro každých 10 minut za den, tzn., že den, tj. 24 hodin, je rozdělen na 144 10-ti minutových částí a pro každou tuto část je uchováván počet vozidel, který za daný časový interval projel parkovištěm či bránou.

U mýtných bran je v každých 10-ti minutách ukládána pouze informace o průjezdu vozidel. Parkoviště však jsou profily rozděleny na více částí. Každé parkoviště tak vede v příslušných 10-ti minutách zvlášť informaci o tom, kolik vozidel v tuto dobu:

- pouze tankovalo za čas nižší než 30 minut,
- parkovalo na dobu odpočinku či stravy na 30 – 90 minut
- parkovalo na denní dobu odpočinku na více než 90 minut.

Aby bylo možné snadněji určit obsazenost parkoviště, je pro každý 10-ti minutový interval u parkoviště dále veden počet vozidel, která byla momentálně na parkovišti přítomna.

Příklad:

- 10:11 – jedno vozidlo, které bude parkovat na dobu 75 minut
- 10:23 – jedno vozidlo, které bude parkovat na dobu více než 4 hodiny
- 10:30 – 10:40 – neprojde parkoviště žádné vozidlo.

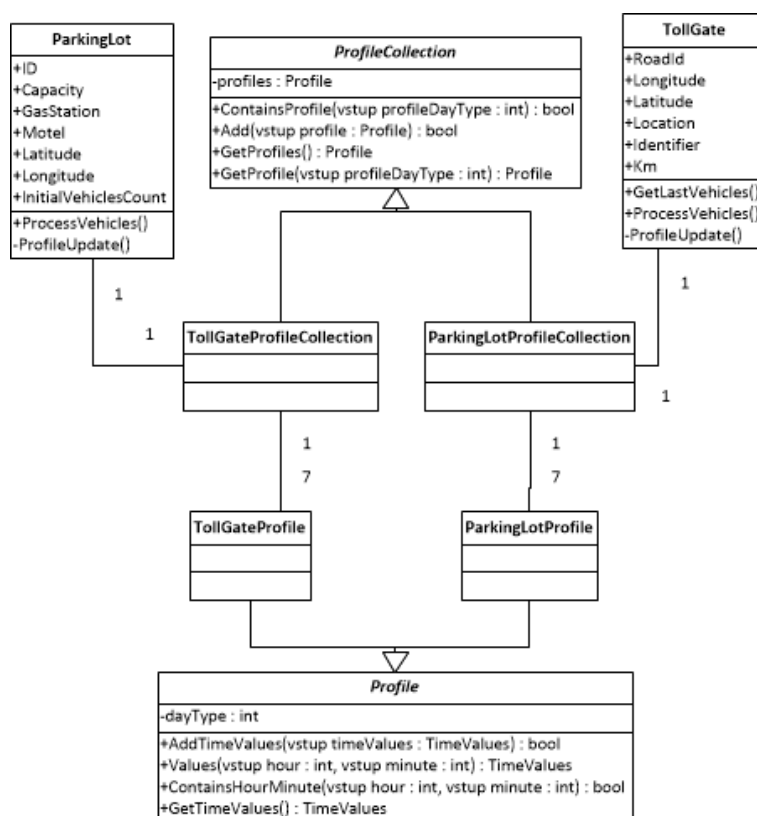
V profilu pro čas 10:30 (časový profil v rozmezí 10:30 – 10:39) není uvedeno ani pro jednu možnost (tankování, stravování, delší odpočinek) žádné vozidlo, ale pro celkový počet vozidel na parkovišti, pokud je splněn předpoklad, že minimálně 15 hodin před 10:11 žádné další vozidlo na parkoviště nepřijelo, jsou uvedena dvě vozidla.

Dále je z analýzy patrné, že pro každé parkoviště není třeba vést profil pro každý den zvlášť, ale pro každý ze sedmi dnů v týdnu vést minimální a maximální hodnotu. To tedy znamená, že pro parkoviště i pro bránu je vytvořeno 7 profilů. Na diagramu 22 je vidět, jakým způsobem jsou profily k parkovištím přiřazeny.

### ProfileCollection

Hlavní kolekce pro držení profilů. Jedná se o abstraktní třídu, která v sobě obsahuje metody pro práci s jednotlivými profily:

- ContainsProfile – jako návratovou hodnotu vrátí, zda se v kolekci nachází profil určený parametrem.
- Add – slouží k přidání nového profilu do kolekce – každá kolekce však může obsahovat nejvýše 7 profilů (pro každý den v týdnu jeden)



Obrázek 22: Třídní diagram profilů bran a parkovišť

- GetProfiles – vrátí všechny profily v kolekci
- GetProfile – vrátí profil pro den určený parametrem

### **ParkingLotProfileCollection**

Jedná se o kolekci profilů pro parkoviště, která dědí z ProfileCollection. Každé parkoviště má jednu tuto kolekci. V každé kolekci je pak 7 profilů – pro každý den v týdnu jeden. Kolekce je v tomto případě opět použita, aby nedošlo k redundanci dat.

### **TollGateProfileCollection**

Stejně jako ParkingLotProfileCollection dědí z ProfileCollection a slouží jako držitel sedmi profilů pro mýtnou bránu.

### **Profile**

Tato třída je hlavním držitelem hodnot profilů. Opět se však jedná o abstraktní třídu, která má metody pro správu hodnot a typ profilu. Typ profilu se řídí podle dne v týdnu, tj. každý den v týdnu je jeden profil a tak i jeden typ. Metody podobě jako u ProfileCollection slouží k udržení neredundantních dat:

- AddTimeValues – přidává časové hodnoty do profilu
- Values – vrací hodnoty počtu vozidel pro časový úsek zadaný hodinou a minutami (minuty nemusí být násobku deseti)
- ContainsHourMinute – vrací informaci, zda jsou v profilu obsaženy hodnoty pro časový úsek zadaný parametry
- GetTimeValues – vrátí pole všech hodnot pro všechny časové úseky v daném profilu

### **TollGateProfile a ParkingLotProfile**

Obě tyto třídy jsou pouze speciálním případem třídy Profile. Liší se pouze ve vnitřním časových hodnotách, které obsahují. TollGateProfile obsahuje časové hodnoty pro mýtnou bránu a ParkingLotProfile obsahuje naopak hodnoty pro parkoviště. Různé druhy časových hodnot jsou uvedeny v dalším diagramu.

### **MinMax**

Třída, která je pouze nosičem dvou celočíselných hodnot. Využívá se pro držení dat pro jeden časový úsek, kde se drží minimální a maximální počet vozidel.



## TimeValues

Abstraktní třída, která slouží jako nositel informace, pro který časový interval jsou dané hodnoty určeny.

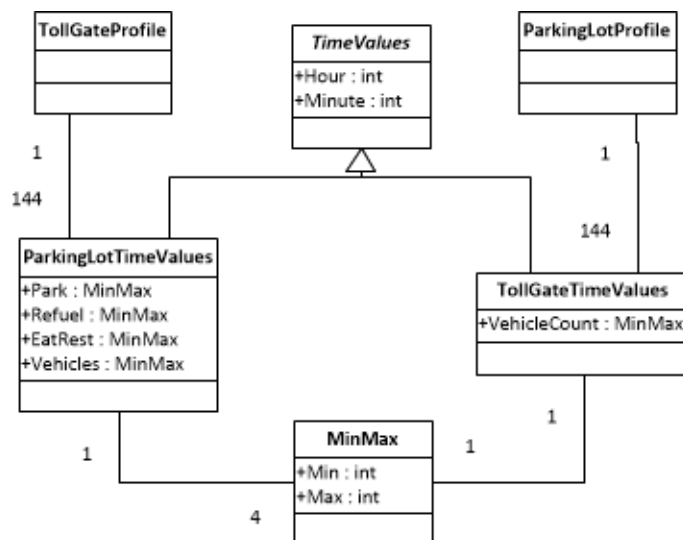
## ParkingLotTimeValues

Potomek třídy TimeValues, která je nosičem MinMax hodnot vozidel pro časový interval na parkovišti. Obsahuje 4 MinMax hodnoty:

- Refuel – kolik vozidel v daný čas tankovalo,
- EatRest – kolik vozidel zastavilo na odpočinek a jídlo,
- Park – kolik vozidel zastavilo na denní dobu odpočinku,
- Vehicles – kolik vozidel bylo na parkovišti přítomno v daném časovém intervalu.

## TollGateTimeValues

Opět se jedná o potomka třídy TimeValues a drží jedinou MinMax proměnnou – VehiclesCount, která slouží pro zaznamenávání počtu vozidel, které projely bránou v daném časovém intervalu.



Obrázek 23: Třídní diagram časových hodnot jednotlivých profilů

### 7.1.7 Aktualizace profilů

U profilů jsou uloženy hodnoty, kolik vozidel splňovalo dané podmínky (dobu strávenou na parkovišti atp.) v daném časovém intervalu. Aby však bylo možné časové intervaly

aktualizovat, musela se nejdříve vyřešit situace, kdy vozidlo už dále nemůže změnit stav nebo může v daném intervalu projet více vozidel.

## Profily mýtných bran

Nejprve se vyřešila aktualizace profilů mýtných bran. Zde je situace jednodušší – stačí pouze sledovat konec časového intervalu a na jeho konci je možné data uložit do databáze.

Pro tento účel je u mýtné brány zavedeno nové pole, které slouží jako dočasný držitel hodnot, které se později uloží do databáze a ze kterých se aktualizuje profil. Toto pole je další kolekce profilů, ale u každého profilu je vyplněna pouze minimální hodnota. To proto, že je potřeba udržovat pouze aktuální průjezdnost.

Další proměnná je časová značka, která říká, kdy naposled došlo k aktualizaci profilu. Je totiž zbytečné aktualizovat vždy celý profil.

Aktualizace se provádí na konci zpracování vozidel v metodě `ProcessVehicles` třídy `TollGate`. Vyvolá se metoda `ProfileUpdate`, jejíž tělo vypadá následovně:

---

```

DateTime tmpTime = DateTime.Now;
DateTime updateTime = _lastProfileUpdate;

if (tmpTime.Subtract(_lastProfileUpdate).TotalMinutes < 10)
    return;

TollGateProfileMapper profileMapper = new TollGateProfileMapper();

do
{
    ProfileDayType dayType = TollGateProfile.GetDayType(updateTime.DayOfWeek);
    TollGateTimeVaues valuesToUpdate =
        _tmpUpdateProfile[dayType][updateTime.Hour, updateTime.Minute];
    TollGateTimeVaues valuesToBeUpdated =
        _profiles [dayType][updateTime.Hour, updateTime.Minute];
    bool update = false;
    if (valuesToBeUpdated.VehicleCount.Min > valuesToUpdate.VehicleCount.Min ||
        valuesToBeUpdated.VehicleCount.Min < 0)
    {
        valuesToBeUpdated.VehicleCount.Min = valuesToUpdate.VehicleCount.Min;
        update = true;
    }
    if (valuesToBeUpdated.VehicleCount.Max < valuesToUpdate.VehicleCount.Min)
    {
        valuesToBeUpdated.VehicleCount.Max =
            valuesToUpdate.VehicleCount.Min;
        update = true;
    }
}
if (update)
{
    profileMapper.
        UpdateProfileTimeValues(_identifier, dayType, valuesToBeUpdated);
}
valuesToUpdate.VehicleCount.Min = 0;
updateTime = updateTime.AddMinutes(10);

```

```
} while (updateTime < tmpTime.AddMinutes(-10) ||  
    ((updateTime < tmpTime && updateTime > tmpTime.AddMinutes(-10))  
    && (int)(updateTime.Minute / 10) != (int)(tmpTime.Minute / 10)));
```

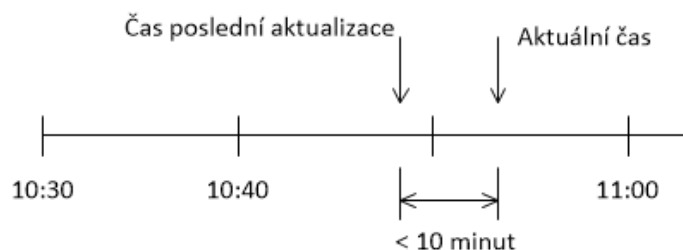
```
_lastProfileUpdate = tmpTime;
```

---

### Výpis 1: Metoda ProfileUpdate

Funkčnost této metody je možné popsat následovně:

1. Uloží se aktuální čas a čas poslední aktualizace profilu.
2. Zjistí se, zda od poslední aktualizace uběhlo více než 10 minut. Pokud ne, není co aktualizovat a aktualizace končí.
3. Pokud ano, zjistí se, který časový interval se bude aktualizovat, a k němu se načtou aktualizací data.
4. Provede se kontrola, zda nově naměřená hodnota spadá do již existujícího intervalu nebo zda je vyšší či nižší než maximální resp. minimální hodnota. U minimální hodnoty je v defaultním profilu uloženo vždy záporné číslo. Pokud by totiž byla jako defaultní hodnota uložena nula a při první aktualizaci by byla naměřena hodnota např. 10, nedošlo by k aktualizaci a data by tak byly zkreslené. U maximální hodnoty tato podmínka není potřeba.
5. Pokud došlo ke změně některé z hodnot, provede se uložení této hodnoty do databáze.
6. Vynuluje se hodnota dočasného profilu pro další zpracování.
7. Posune se časová značka o 10 minut dopředu, aby ukazovala do dalšího časového intervalu.
8. Zkontroluje se, zda se bude aktualizovat další časový interval. Aktualizace dalšího intervalu se provádí od bodu 3 a může nastat za dvou podmínek:
  - (a) mezi aktuálním časem a poslední aktualizací je časový rozestup větší než 10 minut. To znamená, že je zde určitě alespoň jeden volný interval, který je možné aktualizovat.
  - (b) mezi aktuálním časem a časem poslední aktualizace je rozdíl menší než 10 minut, ale každý z časů ukazuje do jiného intervalu 24.
9. Po aktualizování všech intervalů se uloží aktuální čas jako čas poslední aktualizace. Tento čas pak ukazuje na interval, který ještě nebyl aktualizovaný.



Obrázek 24: Příklad aktualizace časového úseku

### Profily parkovišť

U parkovišť se provádí aktualizace profilů po zpracování vozidel na konci metody `ProcessVehicles`. Metoda, která provádí aktualizaci, se nazývá stejně jako u mýtné brány `ProfileUpdate` a stejně jako u mýtné brány, tak i zde jsou zavedeny dvě nové proměnné. Jednou z nich je časová značka poslední aktualizace a druhou je dočasná kolekce profilů parkovišť, do kterých se ukládá opět pouze minimální hodnota jako hodnota vozidel, které splnily podmínky v daném intervalu.

Tělo metody je téměř totožné s tělem metody pro aktualizaci profilů mýtných bran. Liší se v aktualizaci ne jedné, ale čtyř hodnot (`Refuel`, `EatRest`, `Park`, `Vehicles`) a hlavním rozdílem je určení referenčního času pro aktualizaci.

Jako referenční čas se totiž nemůže brát v potaz pouze aktuální čas, ale musí se počítat také s vozidly, které jsou v úseku ještě přítomny. Mohlo by totiž dojít k situaci, kdy se vozidlo započítá do profilu jako tankující, ale za 30 minut, kdyby vozidlo stále úsek neopustilo, by se započítalo znova jako vozidlo, které zastavilo kvůli odpočinku. Problém by také nastal při započítávání vozidel, které sjely některým z exitů. Takže v případě parkovišť, které mají mezi některou z bran a parkovištěm exit, dojde k aktualizaci profilu nejpozději 15 hodin po průjezdu takového vozidla příjezdovou bránou.

Pro určení referenčního času je využit tento algoritmus:

```
Vehicle oldestParkedCar = _parkedVehicles.FirstOrDefault();
Vehicle oldestTmpCar = _tmpVehicles.FirstOrDefault();
DateTime tmpTime = DateTime.Now;

if (oldestParkedCar != null && oldestTmpCar == null &&
    oldestParkedCar.PassingTime < tmpTime &&
    oldestParkedCar.PassingTime > _lastProfileUpdate)
    tmpTime = oldestParkedCar.PassingTime;
else if (oldestParkedCar == null && oldestTmpCar != null &&
    oldestTmpCar.PassingTime < tmpTime &&
    oldestTmpCar.PassingTime > _lastProfileUpdate)
    tmpTime = oldestTmpCar.PassingTime;
else if (oldestParkedCar != null && oldestTmpCar != null)
{
    if (oldestParkedCar.PassingTime < tmpTime)
        tmpTime = oldestParkedCar.PassingTime;
    if (oldestTmpCar.PassingTime < tmpTime)
        tmpTime = oldestTmpCar.PassingTime;
}
```

}

### Výpis 2: Určení referenčního času pro aktualizaci profilu parkoviště

Podmínky v tomto algoritmu je možné popsat následujícím způsobem:

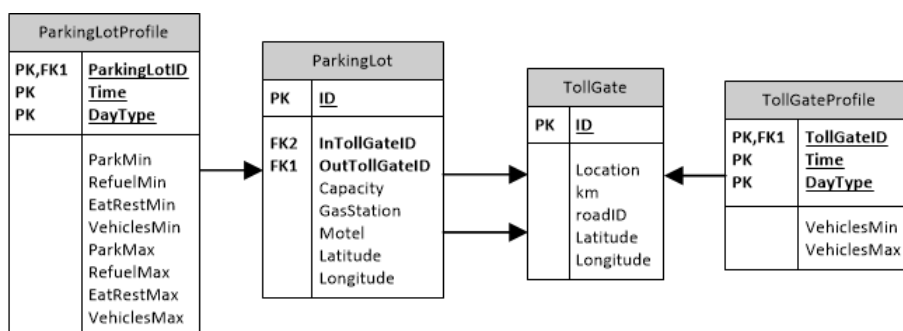
1. Ze seznamu zaparkovaných vozidel a dočasného seznamu (seznamy jsou seřazeny chronologicky dle času příjezdu) se získají z každého první vozidlo.
2. Uloží se aktuální čas jako referenční čas.
3. Pokud není žádné zaparkované vozidlo a existuje alespoň jedno vozidlo v úseku, zjistí se, zda průjezdový čas tohoto vozidla je starší než aktuální čas. V případě, že ano, je čas tohoto vozidla použit jako referenční.
4. Pokud existuje zaparkované vozidlo a neexistuje žádné vozidlo v dočasném seznamu, je porovnán čas zaparkovaného vozidla s aktuálním časem a pokud je tento starší, je použit jako referenční čas.
5. Pokud existují vozidla jak v seznamu zaparkovaných, tak i v dočasném seznamu, jsou jejich časy porovnány s aktuálním časem a je vybrán nejstarší z nich.

Jakmile je vybrán referenční čas, je tento použit ve stejném algoritmu jako u aktualizace profilu mýtné brány, jako aktuální čas.

#### 7.1.8 Uložení dat v databázi

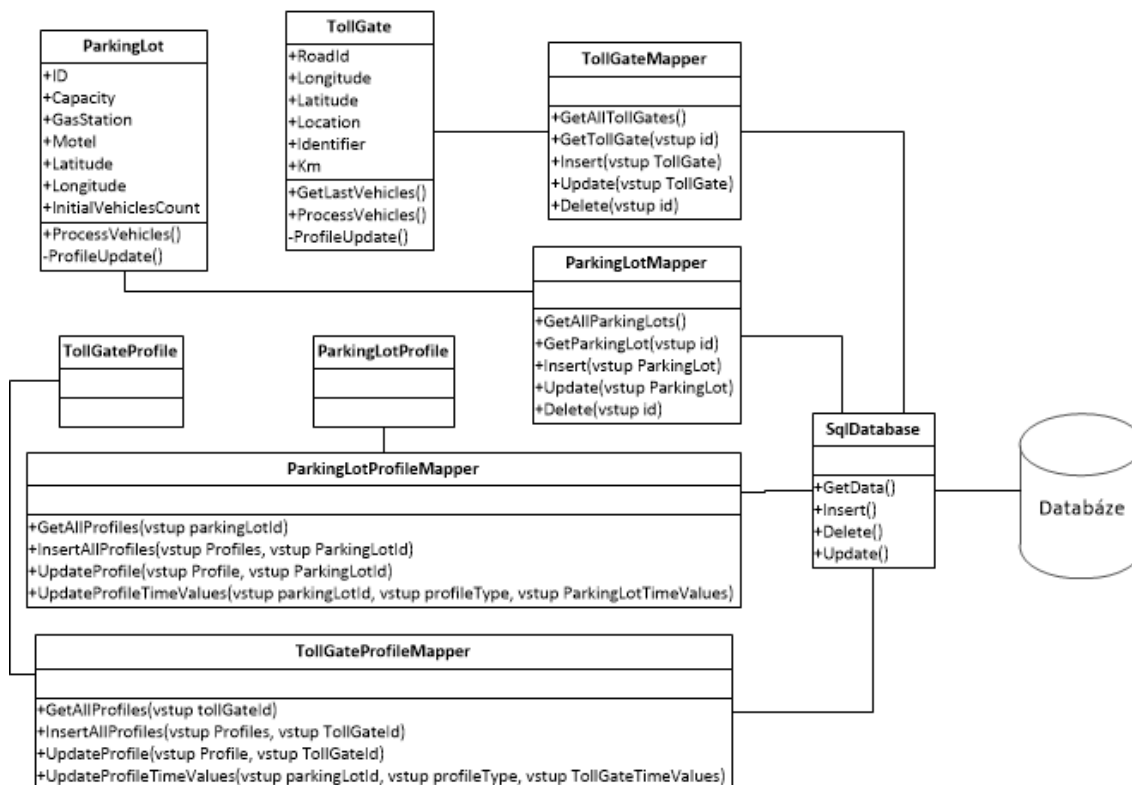
Data z parkovišť, mýtných bran a jejich přidružených profilů je potřeba dlouhodobě ukládat. Jako způsob uložení byla nejdříve zvolena SQLite databáze, kvůli jednoduchosti a možné přenositelnosti bez nutnosti instalace dalšího softwaru. Po první aktualizaci profilů však kvůli rychlosti zápisu byla tato databáze nahrazena technologií od Microsoftu, a sice MS SQL databáze.

Do databáze není potřeba z parkovišť a mýtných bran ukládat velké množství informací. Největší obsah budou mít tabulky profilů, kde pro jedno parkoviště nebo bránu bude uloženo celkem  $7 \times 144 = 1008$  záznamů. Databázový model je znázorněn diagramem.



Obrázek 25: Databázové tabulky pro uložení dat

Aby nebylo nutné přidávat do již existujících tříd další metody, které by se staraly o ukládání dat do databáze, byl použito rozložení podle návrhového vzoru DataMapper [3]. Na základě tohoto vzoru byly vytvořeny mappery, které jsou znázorněné na následujícím diagramu 26. Každý z těchto mapperů má dále informaci jak uložit nebo načíst daný objekt do resp. z databáze.



Obrázek 26: Třídní diagram Data mapperů pro práci s databází

### 7.1.9 Poskytování informací z Parking manageru

Informace o parkovišti je možné zjistit přímo v klientské aplikaci Parking manager. Aby však bylo možné informace poskytovat i mimo tuto aplikaci, muselo být vytvořeno rozhraní, které umožňuje snadno a rychle tyto data získávat. Možností, jak poskytovat data vnějším systémům (ne jenom v rámci jednoho počítače nebo serveru, ale skrze internet), je několik. Buď jsou data ukládána do databáze, nebo do souboru a další systémy, které potřebují dále tyto data zpracovávat, si tyto informace načtou právě z těchto zdrojů.

Poskytování informací skrze další uložení je poměrně náročné na konfiguraci. Musí se totiž zajistit přístup k daným souborům. Snadnější je proto vytvořit jiný typ přístupu. Nejefektivnější je tedy vytvoření webové služby, která bude poskytovat tyto informace v reálném čase a to bez nutnosti ukládání dat do dalších souborů či databází.

Pro tento účel byla do Parking manageru přidána možnost poskytovat informace skrze WCF webovou službu. To zda bude služba aktivní či nikoli je možné ovládat přímo v hlavním okně Parking manageru. Je zde také možnost definovat URL adresu, na které bude služba dostupná.

Webová služba však při standardním nastavení při každém zavolání vytvoří novou instanci objektu. To by v tomto případě bylo nežádoucí. Při každém zavolání některé z metod webové služby by docházelo k opětovnému načítání dat z databáze a objekty by navíc neobsahovaly data o aktuálním provozu. Proto musela být vytvořena služba, která má pouze jednu instanci společnou pro všechny uživatele, kteří službu využívají. WCF webová služba musí mít pro tento účel dvě nastavení.

První nastavení se týká třídy, která implementuje rozhraní služby. Přímou deklaraci třídy musí být uveden atribut `ServiceBehavior` s parametrem `InstanceContextMode = InstanceContextMode.Single`. Tento parametr říká, že bude vytvořena pouze jedna instance, která bude společná pro všechny uživatele.

---

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
public class ParkingLotConnector : IParkingLotJSONConnector
{...
```

---

#### Výpis 3: Aktivace jediné instance objektu pro webovou službu

Druhé nastavení se týká hostovacího kódu, který bude zajišťovat spuštění a dostupnost služby. U standardní služby by se při každém spuštění služby vždy vytvořila nová instance objektu držícího data. V tomto případě je však potřeba do služby poskytnout aktuální data z parkovišť. Toho je možno docílit vytvořením hostu, který přebírá již přímo existující instanci seznamu parkovišť, se kterou dále pracuje.

---

```
ParkingLotConnector _connector = new ParkingLotConnector(_parkingLots);
Uri baseAddress = new Uri(serviceUrlTxt.Text);
ServiceHost _host = new ServiceHost(_connector, baseAddress);
```

---

#### Výpis 4: Přiřazení vytvořené instance objektu do hostu webové služby

Dalším poměrně důležitým problémem je, v jakém formátu budou data poskytována vnějšímu světu. Přímou standard WCF, který zveřejňuje data pomocí SOAP formátu, umožňuje konzumaci služby všem, kteří využívají .NET frameworku. I jiné programovací jazyky jako je Java jsou schopny SOAP formát zpracovat. Problém však nastává i jiných jazyků, kde není tato podpora nativní. Proto bylo rozhodnuto poskytovat data ve dvou dnes nejrozšířenějších formátech a sice JSON a XML.

Aby bylo možné data zveřejňovat ve dvou různých formátech, bylo nutné pro každý z těchto formátů vytvořit zvláštní kontrakt. Každý s jinou sadou metod, které budou vracet data ve formátu XML nebo JSON. Pro každý tento kontrakt (interface) musí u služby vytvořený zvláštní koncový bod (endpoint) s vlastní adresou, který bude určovat, že na této adrese jsou k dispozici data právě v daném formátu.

V následující ukázce je uvedeno řešení, jak softwarově vytvořit dva koncové body, kde každý z nich bude podporovat jiný formát. První z nich bere jako kontrakt `IParkingLotJSONConnector` a výstupní formát odpovědi je ve formátu JSON. Adresa tohoto koncového bodu je json. Obdobně je to i u formátu XML.

```

WebHttpBehavior whpJson = new WebHttpBehavior();
whpJson.DefaultOutgoingResponseFormat = WebMessageFormat.Json;
ServiceEndpoint se =
    _host.AddServiceEndpoint(typeof(IParkingLotJSONConnector),
        new WebHttpBinding(), "json");
se.Behaviors.Add(whpJson);

```

```

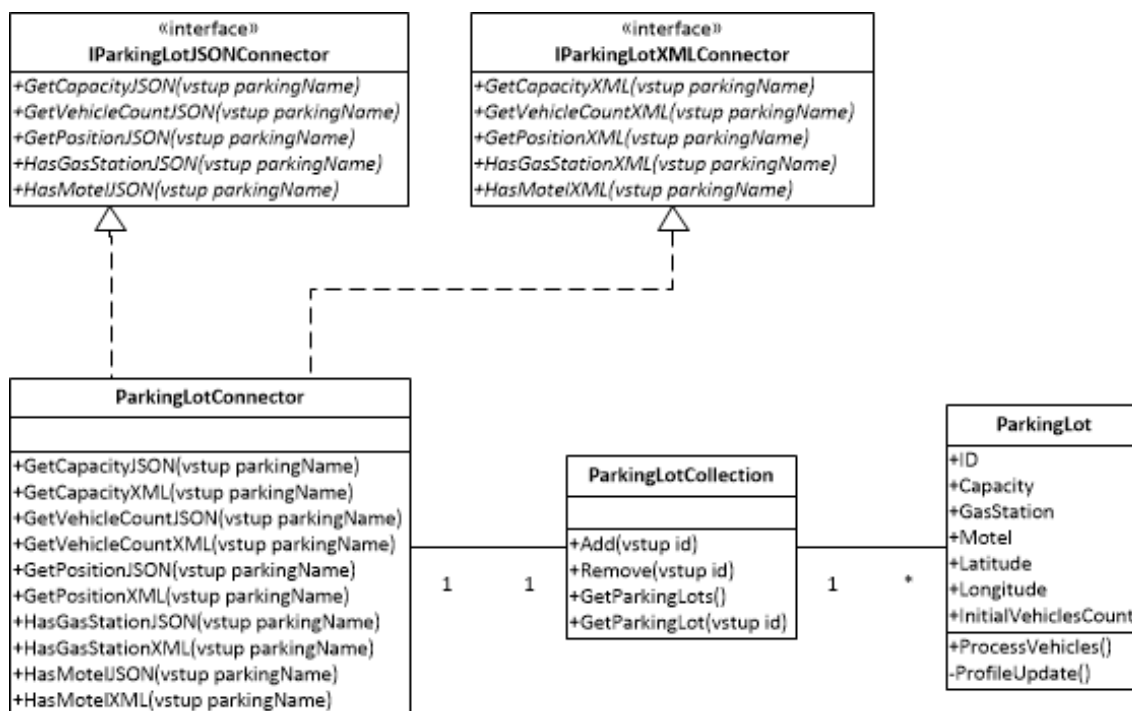
WebHttpBehavior whpXml = new WebHttpBehavior();
whpXml.DefaultOutgoingResponseFormat = WebMessageFormat.Xml;
se = _host.AddServiceEndpoint(typeof(IParkingLotXMLConnector),
    new WebHttpBinding(), "xml");
se.Behaviors.Add(whpXml);

```

Výpis 5: Vytvoření dvou endpointů pro JSON a XML formát

Adresy koncových bodů se pak spojí se základní adresou zadanou v Parking manageru a na výsledné URL adrese jsou pak k dispozici data v daném formátu. Je-li například základní adresa `http://localhost:8484/` a požadovaný formát JSON, adresa pro přístup je pak „`http://localhost:8484/json/metoda`“.

Rozložení a využívané data ve webové službě je znázorněno diagramem na obrázku 27. Rozhraní `IParkingLotJSONConnector` a `IParkingLotXMLConnector` jsou předpisy pro různé formáty výstupních dat. `ParkingLotConnector` pak obsahuje definici všech těchto metod, přičemž tento objekt již má odkaz na instanci kolekce všech parkovišť v systému.



Obrázek 27: Třídní diagram webové služby pro poskytování dat



Informace o daném parkovišti tak lze přes webovou službu získat např. tímto způsobem:

1. Základní adresa služby: `http://localhost:8484/`
2. Požadovaný výstupní formát: XML
3. Zjišťovaný parametr: kapacita parkoviště
4. Sledované parkoviště: P210

Výsledná URL adresa pro získání uvedených dat je následující:

`http://localhost:8484/xml/GetCapacityXML?parkingName=P210`

Služba vrátí data ve formátu XML:

`<int>40</int>`

## 7.2 Gate simulator

Vzhledem k tomu, že bylo potřeba provést zpracování historických dat, nebylo možné Parking manager spustit v reálném čase. Pro účely historických dat a také pro účely urychlení pracování vznikl Gate simulator. Hlavním úkolem tohoto simulátoru je simulace času a jeho zrychlování pro jedno parkoviště.

Simulace času však není jedinou funkcionalitou. Simulátor má několik možností:

1. Zobrazení nezpracovaných vozidel z obou mýtných bran.
2. Zobrazení všech vozidel, která jsou momentálně přítomna v úseku mezi oběma branami.
3. Zobrazení vozidel, která jsou v dočasném seznamu parkoviště.
4. Zobrazení vozidel, která jsou na parkovišti zaparkovaná.
5. Vypnout aktualizaci některého z profilů – parkoviště, příjezdová nebo odjezdová mýtná brána.
6. Nastavení, zda se profily příjezdové mýtné brány budou nastavovat automaticky z uložených profilů nebo ručně pomocí nastavení minimálního a maximálního počtu vozidel.
7. Nastavení parkoviště, zda se počty vozidel budou brát z profilů nebo se budou nastavovat ručně pomocí minimálního a maximálního počtu pro každou možnost (Refuel, EatRest a Park).
8. Nastavení, zda se budou používat ostrá data, která jsou k dispozici jako historická data v souborech nebo se bude používat ruční nastavení distribuce vozidel.
9. Nastavení rychlosti, minimální rychlosti, která se využívá při výpočtu času potřebného na projetí od jedné brány k druhé.

### 7.2.1 Simulace času

Simulátor má možnost simulovat čas a to nejen nastavení začátku, ale také nastavení rychlosti. Je tedy možné zvýšit rychlost běhu času až několik tisíc krát rychleji. Vše záleží pouze na výkonu zařízení, na kterém je simulátor spuštěn. Dále je možné tok času zastavit a opět spustit.

Aby bylo však možné parkovišti i mýtným branám vnutit jiný čas než aktuální, bylo nutné do obou těchto tříd přidat další proměnnou, která v sobě nese informaci o simulovaném čase. Tento simulovaný čas je pak nutné nastavit před každým zavoláním metod `ProcessVehicles` jak u bran, tak i u parkoviště.

---

```
_parkingLot.CurrentTime = currentTime;
_parkingLot.ProcessVehicles();
```

---

Výpis 6: Nastavení simulátorového času parkovišti

Dále se ve třídách `ParkingLot` a `TollGate` musely upravit všechny místa, kde se načítal aktuální čas a to kódem, který zjišťuje, zda je nastavený simulovaný čas či nikoli a podle toho se rozhoduje, který z časů použít.

---

```
DateTime tmpTime =
    (_currentTime != DateTime.MinValue) ? _currentTime : DateTime.Now;
```

---

Výpis 7: Použití simulátorového času místo času aktuálního

Aktualizaci simulovaného času má na starosti hlavní smyčka. Jedná se o metodu, která je spuštěna v novém vlákne a uvnitř této metody je spuštěn nekonečný cyklus `while`. Tento cyklus je rozdělený na dvě části.

První část má za úkol spuštění zpracování parkovišť, mýtných bran a nastavení počtu vozidel. Toto je řešeno vnitřním cyklem, který proběhne tolikrát, jak je nastavená rychlost toku času. Pokud je tedy rychlost nastavená na 1 (reálný čas), provede se vnitřní cyklus pouze jednou a čeká se jednu vteřinu (ne celou, kalibruje se podle doby trvání zpracování). Pokud je nastavená rychlost např. 1000, vnitřní cyklus se provede 1000x a poté se čeká jednu vteřinu do dalšího spuštění.

V tomto cyklu se provede každých 10 minut nastavení hodnot z profilů (případně vygenerování vozidel), každou celou minutu se spustí kalkulace vozidel mýtných bran a každých 5 minut kalkulace vozidel parkovišť.

---

```
for (int i = 0; i < _speed; i++)
{
    if (_time.Minute % 10 == 0 && _time.Second == 0)
        AdjustProfilesAndGenerateVehicles(_time);
    if (_time.Second == 0)
        DoTollGatesCalculation(_time);
    if (_time.Minute % 5 == 0 && _time.Second == 0)
        DoParkingCalculation(_time);
    _time = _time.AddSeconds(1);
}
```

---

Výpis 8: Cyklus volající kalkulační metody

Druhá část nekonečného cyklu je řídicí. Tato část má za úkol nekonečný cyklus pozastavit, zastavit nebo nechat běžet. Pro řízení se využívá příznak `_simulatorNextState`, který říká, jaká má být následující akce simulátoru.

Pokud se má pozastavit, využívá se vláknového pozastavení pomocí `Thread.Sleep(Timeout.Infinite)`, který vlákno zastaví na dobu neurčitou. Později je možné vlákno opět zprovoznit pomocí funkce `_simulatorThread.Interrupt()`.

Ukončení se řeší jednoduchým zastavením nekonečného cyklu, a pokud není nastaveno ani zastavení ani pozastavení, cyklus pokračuje dále tak, že počká maximálně jednu vteřinu (jinak od jedné vteřiny odečte čas, který byl potřeba na zpracování všech výpočtů) a pak pokračuje od začátku.

---

```

switch (_simulatorNextState)
{
    case SimulatorNextStates.Pause:
        _simulatorState = SimulatorStates.Paused;
        try
        { Thread.Sleep(Timeout.Infinite); }
        catch (ThreadInterruptedException)
        {
            if (_simulatorNextState == SimulatorNextStates.Stop)
            {
                _simulatorState = SimulatorStates.Stoped;
                run = false;
            }
            else
                _simulatorState = SimulatorStates.Running;
        }
        break;
    case SimulatorNextStates.Stop:
        run = false;
        _simulatorState = SimulatorStates.Stoped;
        break;
    case SimulatorNextStates.Run:
        _simulatorState = SimulatorStates.Running;
        long ticksAfter = DateTime.UtcNow.Ticks;
        double correction = (ticksAfter - ticksBefore) / 10000;
        correction = (correction >= 1000) ? 1 : correction;
        Thread.Sleep((int)(1000 - correction));
        break;
}

```

---

Výpis 9: Řízení běhu simulátoru

## 7.2.2 Zobrazování vozidel

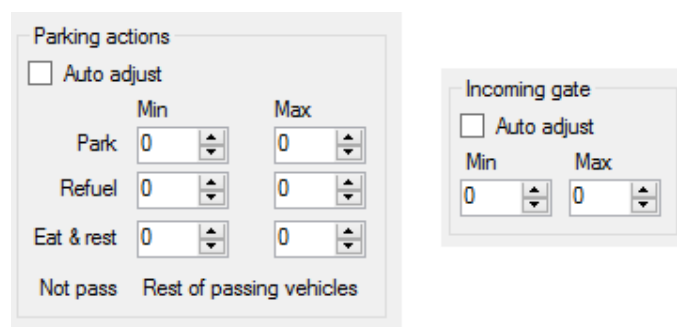
Zobrazení všech výše uvedených vozidel se provádí opět v nekonečném cyklu, který je podobně jako hlavní smyčka řízen příznakem stavu simulátoru. Aktualizace zobrazovaných dat se provádí každou vteřinu reálného času a jsou vypisovány tyto informace:

- Zobrazení nezpracovaných vozidel mýtných bran je ve dvou seznamech, kde každý z nich se plní daty ze seznamu vozidel, která jsou momentálně v seznamu nezpracovaných u obou bran.
- Seznam všech vozidla mezi branami zobrazuje vozidla, která byla vygenerována simulátorem.
- Zobrazení vozidel, která jsou v dočasném seznamu parkoviště a vozidel, která jsou již zaparkovaná, se provádí každé do zvláštního seznamu.

### 7.2.3 Simulování průjezdu vozidel

Jakmile je v simulátoru aktivována možnost používání reálných dat, nepoužívá se žádné další nastavování hodnot. Zavolají se pouze postupně metody ProcessVehicles() nejprve na příjezdové bráně, poté na odjezdové a nakonec na parkovišti. Tím dojde ke zpracování dat ze vstupních souborů s informacemi přímo z mýtných bran.

Pokud však tato možnost vybraná není, přistupuje se ke generování vozidel na základě buď hodnot nastavených v ovládacích prvcích simulátoru, nebo se hodnoty použijí z profilů příjezdové mýtné brány a parkoviště. Generování vozidel se provádí vždy pro jeden časový úsek 10-ti minut.



Obrázek 28: Manuální nastavení profilů v simulátoru

V případě, že je tedy vybraná možnost, že se mají používat hodnoty z profilů, jsou tyto hodnoty načteny do globálních proměnných. Používají se pouze hodnoty z profilů, které podle simulovaného času odpovídají vybraným intervalům. V opačném případě jsou hodnoty nastaveny již během nastavování v ovládacích prvcích v GUI simulátoru 28.

```

if (_useAutoAdjustParkingLot)
    parkingLotProfile =
        _parkingLot.Profiles [ ParkingLotProfile.GetDayType(referenceTime.DayOfWeek)];
if (_useAutoAdjustInGate)
    tollGateProfile =
        _parkingLot.IncomingTollGate.
        Profiles [ TollGateProfile.GetDayType(referenceTime.DayOfWeek)];

```

---

```
// set values from profile if auto adjust is enabled
if (_useAutoAdjustInGate && tollGateProfile != null)
{
    MinMax values =
    tollGateProfile [referenceTime.Hour, referenceTime.Minute].VehicleCount;
    _inGateVehiclesMin = (values.Min < 0) ? 0 : values.Min;
    _inGateVehiclesMax = (values.Max < 0) ? 0 : values.Max;
}
if (_useAutoAdjustParkingLot && parkingLotProfile != null)
{
    ParkingLotTimeValues values =
    parkingLotProfile [referenceTime.Hour, referenceTime.Minute];
    _eatRestMin = (values.EatRest.Min < 0) ? 0 : values.EatRest.Min;
    _eatRestMax = (values.EatRest.Max < 0) ? 0 : values.EatRest.Max;
    _refuelMin = (values.Refuel.Min < 0) ? 0 : values.Refuel.Min;
    _refuelMax = (values.Refuel.Max < 0) ? 0 : values.Refuel.Max;
    _parkMin = (values.Park.Min < 0) ? 0 : values.Park.Min;
    _parkMax = (values.Park.Max < 0) ? 0 : values.Park.Max;
}
```

---

#### Výpis 10: Využití manuálně nastavených hodnot profilů

Jakmile jsou hodnoty nastaveny, je možné přistoupit k samotnému generování vozidel, které se skládá z několika kroků:

1. Z intervalů z profilů (nebo ze zadaných hodnot) se náhodně vyberou hodnoty pro počet vozidel na příjezdové bráně, počet parkujících, tankujících a odpočívajících vozidel.
2. Výpočet intervalu minimálního rozestupu vozidel.
3. Vygenerování vozidel a jejich náhodně rovnoměrná distribuce do intervalu 10-ti minut.
4. Přiřazení statusu (Parked, EatRest, Refuel, NoPass) vygenerovaným vozidlům.

Náhodné vybrání hodnot je provedeno pomocí objektu Random pouze pro počet vozidel z mýtné brány, parkujících, tankujících a odpočívajících vozidel. Vozidla, která nebudou parkovištěm projíždět, jsou dopočítána jako rozdíl počtu vozidel z brány a součtu všech vozidel náhodně určených na parkovišti.

---

```
int vehiclesCount = _rand.Next(_inGateVehiclesMin, _inGateVehiclesMax);
int parkCount = _rand.Next(_parkMin, _parkMax);
int refuelCount = _rand.Next(_refuelMin, _refuelMax);
int eatRestCount = _rand.Next(_eatRestMin, _eatRestMax);
int noPassCount = vehiclesCount - parkCount - refuelCount - eatRestCount;
```

---

#### Výpis 11: Náhodné určení počtů vozidel pro parkoviště a bránu

Z počtu vozidel na příjezdové bráně se dále vypočte, na kolik dílů se rozdělí 10-ti minutový interval. Například při počtu 5 vozidel, se rozdělí na intervaly po dvou minutách. Tím se zajistí, aby vozidla byla rozdělena do časového intervalu rovnoměrně a

aby nedošlo k situaci, že by všechny vozidla měla stejný příjezdový čas. To je totiž téměř nepravděpodobné, ale hlavně technicky neproveditelné. V jednom okamžiku totiž bránou můžou projet nanejvýš dvě vozidla u silnice, která má v jednom směru právě dva jízdní pruhy.

---

```
List<Vehicle> vehicles = new List<Vehicle>();
DateTime start = referenceTime;
int interval = 600000 / ((vehiclesCount == 0) ? 1 : vehiclesCount);
```

---

#### Výpis 12: Určení intervalu pro generovaná vozidla

Po určení počtu intervalů uvnitř jednoho 10-ti minutového úseku je potřeba vygenerovat samotná vozidla. Počet vozidel se řídí počtem vozidel z příjezdové brány. Příjezdový čas je pak určen tak, že v rámci podintervalů je ještě náhodně vygenerován čas, kde se vozidlo nachází. Tím je zajištěno, že vozidla nebudou mít od sebe pevné časové rozestupy, ale jejich rozestupy se mohou lišit.

---

```
for (int i = 0; i < vehiclesCount; i++)
{
    int milliseconds = _rand.Next(0, interval + 1);
    vehicles.Add(new Vehicle(_idCounter++.ToString(),
        start.AddMilliseconds(milliseconds)));
    start = start.AddMilliseconds(interval);
}
```

---

#### Výpis 13: Generování vozidel s náhodným časovým rozestupem

Jakmile jsou již vygenerována všechna vozidla a mají určen čas příjezdu, zbývá už jim pouze přiřadit status a tím i čas, jak dlouho na parkoviště stráví (případně přes parkoviště vůbec neprojedou). Protože však třída `Vehicle` neobsahuje informaci o tom, jak dlouhou dobu vozidlo na parkovišti stráví, ale jenom čas příjezdu a čas odjezdu, který se ale zpracovává až u zpracování parkovištěm, bylo třeba vytvořit kontejner, který v sobě nese jak samotné vozidlo (třída `Vehicle`), tak status vozidla a čas, jak dlouho se na parkovišti zdrží. Tímto kontejnerem je třída `SimulatorVehicle`.

Seznam vozidel pro simulátor je vytvořen tak, že se z již vygenerovaných vozidel náhodně vybírají vozidla a těm se přiřadí některý ze zbývajících stavů. Všem vozidlům, na které již stav nezbyl, se přiřadí stav, že parkoviště vůbec neprojedou. Podle přiřazeného stavu se v konstruktoru `SimulatorVehicle` vygeneruje náhodně čas, jak dlouho bude vozidlo na parkoviště parkovat. Takto vytvořený seznam simulovaných vozidel je dále využit pro zpracování mýtnými branami.

---

```
int cycles = vehicles.Count;
for (int i = 0; i < cycles; i++)
{
    SimulatorVehicle.StayTypes stayType = SimulatorVehicle.StayTypes.NoPass;
    if (parkCount > 0)
    {
        stayType = SimulatorVehicle.StayTypes.Park;
        parkCount--;
    }
}
```

---

---

```

    else if (refuelCount > 0)
    {
        stayType = SimulatorVehicle.StayTypes.Refuel;
        refuelCount--;
    }
    else if (eatRestCount > 0)
    {
        stayType = SimulatorVehicle.StayTypes.EatRest;
        eatRestCount--;
    }
    else
    {
        noPassCount--;
    }
    int index = _rand.Next(0, vehicles.Count);
    _vehiclesGeneratedForProfile.
Add(new SimulatorVehicle(vehicles[index], stayType, _rand));
    vehicles.RemoveAt(index);
}

```

---

Výpis 14: Přiřazení statusu vygenerovaným vozidlům

#### 7.2.4 Využití vygenerovaných vozidel

Běžně mýtné brány využívají ostrá data. Bylo však nutné přidat možnost zpracování vygenerovaných dat do třídy TollGate. Tento problém byl vyřešen přidáním nového parametru do metody ProcessVehicles třídy TollGate. Tento parametr může být prázdný, v tom případě se použije standardní postup načítání dat ze souboru, nebo může být v tomto parametru seznam vygenerovaných vozidel a pak se tento seznam použije místo načítání dat ze souboru.

Seznam vygenerovaných vozidel, který se bude posílat do zpracování mýtnou bránou je částí seznamu, který byl vytvořen při generování vozidel pro profil. Z tohoto seznamu se však vyberou pouze vozidla, která mají plánovaný čas příjezdu menší než je simulovaný čas.

---

```

List<Vehicle> tmpList = new List<Vehicle>();
int cycleCount = _vehiclesGeneratedForProfile.Count;
for (int i = 0; i < cycleCount; i++)
{
    SimulatorVehicle sv = _vehiclesGeneratedForProfile[0];
    if (sv.Vehicle.PassingTime > referenceTime)
        break;

    tmpList.Add(sv.Vehicle);
    _vehiclesBetweenGates.Add(sv);
    _vehiclesGeneratedForProfile.RemoveAt(0);
}
_incomingGate.CurrentTime = referenceTime;
_incomingGate.ProcessVehicles(tmpList, _updateInGateProfile);

```

---

Výpis 15: Výběr vozidel pro zpracování příjezdovou bránou

Pro odjezdovou mýtnou bránu se v případě použití vygenerovaných vozidel nejprve vypočte čas potřebný k projetí úseku mezi branami. Poté se projdou všechny vozidla, která jsou na úseku přítomna a z času předpokládaného odjezdu, simulovaného času a času potřebnému k projetí úseku, se vytvoří seznam vozidel, které musí být zpracovány odjezdovou bránou. Tento seznam se opět předá jako parametr, stejně jako u příjezdové brány.

---

```

float minimalSpeed = _parkingLot.Speed;
DateTime tmpTime = referenceTime.AddSeconds((distance / minimalSpeed * 3600F) * -1.0);
List<SimulatorVehicle> tmpVehicles = new List<SimulatorVehicle>();
foreach (SimulatorVehicle item in _vehiclesBetweenGates)
{
    DateTime probableLeaveTime = item.Vehicle.PassingTime.Add(item.StayDuration);
    if (probableLeaveTime.CompareTo(tmpTime) <= 0)
        tmpVehicles.Add(item);
}
if (tmpVehicles.Count == 0)
    return;
for (int i = 0; i < tmpVehicles.Count; i++)
{
    SimulatorVehicle vsOut = tmpVehicles[i];
    Vehicle vOut = vsOut.Vehicle;
    tmpList.Add(new Vehicle(vOut.ID, tmpTime));
    _vehiclesBetweenGates.Remove(vsOut);
}

_outgoingGate.CurrentTime = referenceTime;
_outgoingGate.ProcessVehicles(tmpList, _updateOutGateProfile);

```

---

Výpis 16: Výběr vozidel pro zpracování odjezdovou bránou

Data pro parkoviště není potřeba nijak zvláště generovat. Parkoviště používá data z mýtných bran a tím je celý systém simulování průjezdů vozidel uzavřen.

### 7.2.5 Aktualizace profilů parkoviště a mýtných bran

Aplikace Gate simulator má možnost vypnutí aktualizování profilů v databázi. Je možné vypnout aktualizaci všech prvků, tj. parkoviště, příjezdové a odjezdové brány zvlášť. To, aby se profily v databázi neaktualizovaly, muselo být provedeno úpravou metod ProcessVehicles. Do obou těchto metod jak u třídy TollGate, tak i u třídy ParkingLot byl přidán nový parametr, pomocí něhož je specifikováno, zda se mají profily aktualizovat, či nikoli. V těle obou metod je tato změna zavedena pouze do podmínky, která kontroluje nastavení nového parametru a podle toho provede nebo neprovede aktualizaci.

### 7.2.6 Spuštění simulátoru

Simulátor má dvě možnosti spuštění. Buď se aplikace spustí samostatně, v tom případě se využije fiktivního parkoviště a dvou fiktivních bran, které však nemají možnost napojení na ostrá data, nebo je možné simulátor spustit přímo z aplikace Parking manager. V



Parking manageru se po vybrání parkoviště zobrazí data o tomto parkovišti. Součástí dat je také tlačítko, které spustí simulátor a předá mu informace o parkovišti a jeho mýtných branách. Poté je možné využít v simulátoru možnost ostrých dat.

### 7.3 Získávání dat ze senzorů

Protože není snadné párovat vozidla z mýtných bran s vozidly ze senzorů, kvůli proměnné délky, nebyla funkcionality načítání a zpracovávání dat ze senzorů naimplementována přímo do Parking manageru. Společností CDV však byla dodána aplikace, která načítání těchto dat řeší.

V aplikaci není pouze holé načítání dat z jednotlivých senzorů. Je zde hned řada dalších funkcí:

- Výběr vstupního a výstupního senzoru.
- Určení časového rozmezí, ze kterého mají být data načteny.
- Specifikace délkových kategorií – pro tento případ byly kategorie určeny 0m – 9m jako osobní automobil a nad 9m jako nákladní vozidlo.
- Možnost aplikovat na vstupní data žádný, jeden nebo všechny vyhlazovací algoritmy.

#### 7.3.1 Převod na časové úseky

Jakmile tedy jsou načteny data ze senzorů, jsou rozřazeny do dvou kategorií – osobní a nákladní vozidla. Poté jsou na tyto kategorie aplikovány všechny vyhlazovací algoritmy. Následně je možné již pročištěná data uložit do CSV souboru.

V datech jsou však stále osobní vozidla a data nejsou rozdělena do časových úseků po 10-ti minutách. Proto byla vytvořena jednoduchá aplikace, která tato data ještě dále zpracovává. Nejprve se tedy načtou do paměti pouze ta vozidla, jejichž délka je větší než 9m. Pak je potřeba roz distribuovat jednotlivé vozidla do profilových časových intervalů (viz. algoritmus 17). Nakonec jsou data profilů uložena do CSV souboru, kde jednotlivé řádky představují časový úsek a sloupce znázorňují jednotlivé dny. Protože z mýtných bran byly k dispozici pouze data ze čtyřech dní, je i u senzorů zkoumán rozsah stejných dní a tyto jsou uloženy do souboru.

---

```
TollGateProfileCollection profiles = new TollGateProfileCollection(0);
foreach (var item in vehicles)
{
    profiles [ TollGateProfile . GetDayType(item.PassingTime.DayOfWeek)]
    [item.PassingTime.Hour, item.PassingTime.Minute].VehicleCount.Min++;
}
```

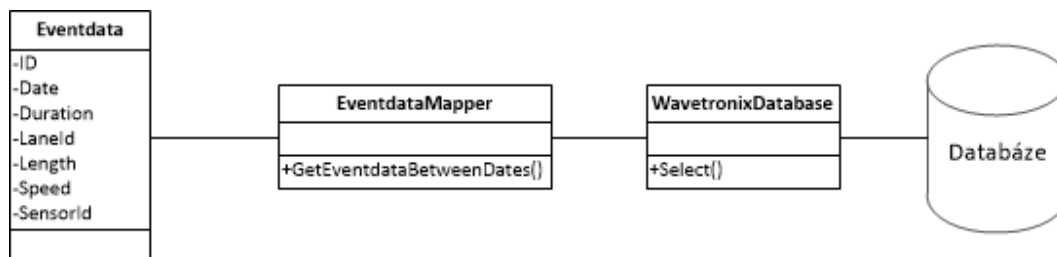
---

Výpis 17: Rozřazení dat do časových úseků

### 7.3.2 Propojení k databázi senzorů

Pro případ budoucího zpracování dat ze senzorů pomocí Parking manageru však bylo vytvořeno napojení na databázi, ve které jsou tato data uložena.

Jedná se o PostgreSQL databázi, ve které je jedna hlavní tabulka – Eventdata. Ta obsahuje stejná data, která načítá dříve zmiňovaná aplikace. Opět i zde je použit návrhový vzor DataMapper, ale tentokrát ve spojení s Data Transfer Object (DTO), kdy je jako datový objekt použit Eventdata. Propojení je patrné na diagramu.



Obrázek 29: Třídní diagram propojení na Wavetronix databázi

## **8 Srovnání výsledků z mýtných bran a senzorů**

Výsledky testů běhu aplikace nad reálnými daty spolu se srovnáním s daty ze senzorů jsou v neveřejné části této práce. Jsou zde uvedeny výsledky z běhu simulátoru pro reálná data z parkovišť PP001 a PP101. Tyto výsledky jsou dále srovnány s výsledky ze senzorů instalovaných na těchto parkovištích.

## 9 Závěr

Z prozkoumání parkovišť na dálnici D1 a z jejich klasifikace vůči mýtným branám je jasné patrné, že pouze některé z nich mají ideální rozložení, tj. takové rozložení, kdy mezi branami a parkovištěm již dále není žádný nájezd ani exit. V případech, kdy jsou v úseku mezi mýtnými branami exity a nájezdy totiž není možné určit přesný počet vozidel, která se na parkovišti zastavila.

Analýza dat z bran a ze senzorů ukázala, že senzory nemají možnost párování vozidel na příjezdu a výjezdu a není tak možné určit, jak dlouho se dané vozidlo na parkovišti zdrželo. Naopak mýtné brány tuto vlastnost mají a je tak možné určit čas, jak dlouho vozidlo strávilo v úseku a tím pádem i na parkovišti. Tento údaj také posloužil pro zpřesnění počtu vozidel na parkovišti tím, že se z parkoviště vyřazovala vozidla, která byla v úseku déle než 15 hodin. Tato doba byla určena několika pokusy nad ostrými daty jako dostatečná. Žádné z vozidel již nebylo v úseku delší dobu.

Byla také definován způsob, jakým je možné ukládat časové profily pro parkoviště a mýtné brány do databáze pro pozdější zpracování. Tyto profily obsahují profil pro každý den v týdnu (celkem tedy 7 profilů pro bránu a 7 profilů pro parkoviště) rozdělení dne na časové intervaly po 10-ti minutách, kde v každém časovém intervalu je uložen počet vozidel, který bránou nebo parkovištěm projel.

Dále byla vytvořena aplikace, která spravuje data z parkovišť a mýtných bran. Tato aplikace má možnost spuštění webové služby, skrze kterou je možné data z parkovišť a bran poskytovat dalším subjektům.

Pro účely zpracování historických dat byl vytvořen simulátor, který má za úkol simulovat běh času, včetně jeho zrychlení. Další možností simulátoru je napojení na reálná data nebo naopak generování fiktivních dat na základě profilů, které byly již dříve zaznamenány z ostrých dat.

Z výsledků, uvedených v neveřejné části práce, které ukazují odchylky dat mýtných bran od dat ze senzorů, je tedy možné usuzovat, že pokud by byl model vyhlazování vozidel, které opustily úsek na některém z exitů, ještě dále rozšířen a zefektivněn, nebo by bylo použito např. doplňkové měření, je možné dosáhnout ještě vyšší efektivity a mýtný systém by tak bylo možné využít jako plnohodnotný monitorovací systém také pro obsazenost dálničních parkovišť.

Dále by se pro účely určení volného parkovacího místa mohla rozšířit a naimplementovat metoda predikce obsazenosti z historických dat. Tím by měl řidič alespoň vyšší jistotu, že parkovací místo zůstane volné i v době, kdy na parkoviště dorazí on sám.

## 10 Reference

- [1] Autoškola Jiří Myšák, *Dopravní značky*, (Online 18.4.2014)  
<http://www.autoskolapodorlicko.cz/znacky/dz-ip-info.html>
- [2] EUR-Lex, *Nařízení komise v přenesené pravomoci (EU) č. 885/2013*, (Online 18.4.2014)  
<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2013:247:0001:0005:CS:PDF>
- [3] Fowler, Martin, *Patterns of Enterprise Application Architecture*, Addison Wesley, 2002
- [4] Hospodářské Noviny, *Dopravní značky*, (Online 18.4.2014)  
[http://multimedia.ihned.cz/?article%5Bgallery%5D%5Bdetail.id%5D=1221240&article%5Bgallery%5D%5Bfrom%5D=&article%5Bgallery%5D%5Bid%5D=934580&article%5Bid%5D=48294250&p=0MF00o\\_d](http://multimedia.ihned.cz/?article%5Bgallery%5D%5Bdetail.id%5D=1221240&article%5Bgallery%5D%5Bfrom%5D=&article%5Bgallery%5D%5Bid%5D=934580&article%5Bid%5D=48294250&p=0MF00o_d)
- [5] Kapsch, *HGV On-Board Unit*, (Online 18.4.2014)  
<http://www.kapsch.net/ktc/downloads/datasheets/in-vehicle/5-8/Kapsch-KTC-DS-OB-4021-01K-EN-WEB?lang=en-US>
- [6] Kapsch, *Výběr mýtného*, (Online 18.4.2014)  
<http://www.kapsch.net/cz/kts/its-solutions/toll-collection>
- [7] MDČR, *Nařízení Evropského parlamentu a rady (ES) č. 561/2006*, (Online 18.4.2014)  
<http://www.mdcr.cz/NR/rdonlyres/8861CCFB-1749-49EC-B007-D215B0AFEFD8/0/5612006.pdf>
- [8] MDČR, *Zákon č. 13/1997 Sb., o pozemních komunikacích*, (Online 18.4.2014)  
[http://www.mdcr.cz/NR/rdonlyres/930E6DB1-40F0-44CC-85D9-04E1A874BFFB/0/ZAKON\\_O\\_POZEMNICH\\_KOMUNIKACICH.13\\_1997\\_ZNENI\\_20130225.rtf](http://www.mdcr.cz/NR/rdonlyres/930E6DB1-40F0-44CC-85D9-04E1A874BFFB/0/ZAKON_O_POZEMNICH_KOMUNIKACICH.13_1997_ZNENI_20130225.rtf)
- [9] MDČR, *Zákon č. 361/2000 Sb., o provozu na pozemních komunikacích*, (Online 18.4.2014)  
<http://www.mdcr.cz/NR/rdonlyres/4AF5FE2D-C52A-49A1-AFF3-49FACD561805/0/3612000Sb.pdf>
- [10] MPSV, *Nařízení vlády č. 590/2006 Sb.*, (Online 18.4.2014)  
[http://www.mpsv.cz/files/clanky/3303/NV\\_prekazky.pdf](http://www.mpsv.cz/files/clanky/3303/NV_prekazky.pdf)
- [11] MVČR, *Jednotný systém dopravních informací*, (Online 18.4.2014)  
<http://www.mvcr.cz/clanek/jednotny-system-dopravnich-informaci.aspx>
- [12] Myto CZ, *Mýtný systém*, (Online 18.4.2014)  
<http://www.myto.cz/cs/novy-uzivatel/mytny-system-1/index.html>
- [13] Myto CZ *Zpoplatněné úseky*, (Online 18.4.2014)  
[http://www.myto.cz/files/files/tariffs/MYTOCZ\\_341\\_tariff\\_table.pdf](http://www.myto.cz/files/files/tariffs/MYTOCZ_341_tariff_table.pdf)

- 
- [14] RITA, *Dedicated Short Range Communications*, (Online 18.4.2014)  
<http://www.its.dot.gov/DSRC/>
- [15] Roman Všečetka, *Mýtná brána pozná i vaše SPZ*, (Online 18.4.2014)  
[http://technet.idnes.cz/mytna-brana-pozna-i-vase-spz-d1p-/tec.technika.aspx?c=A061013\\_121519\\_tec\\_technika\\_vse](http://technet.idnes.cz/mytna-brana-pozna-i-vase-spz-d1p-/tec.technika.aspx?c=A061013_121519_tec_technika_vse)
- [16] Ředitelství silnic a dálnic ČR, *Elektronické mýto*, (Online 18.4.2014)  
<http://dopravniinfo.cz/elektronicke-myto>
- [17] Ředitelství silnic a dálnic ČR, *Národní dopravní informační centrum*, (Online 18.4.2014)  
<http://www.dopravniinfo.cz/narodni-dopravni-informacni-centrum>
- [18] Ředitelství silnic a dálnic ČR, *Odběr dopravních informací*, (Online 18.4.2014)  
<http://www.dopravniinfo.cz/odber-dopravnich-informaci>
- [19] Wavetronix, *Smart sensor HD*, (Online 18.4.2014)  
[http://www.wavetronix.com/uploads/download/file/391/smartsensor\\_hd\\_user\\_guide-20140109092654.en.pdf](http://www.wavetronix.com/uploads/download/file/391/smartsensor_hd_user_guide-20140109092654.en.pdf)
- [20] Wavetronix, *Smart sensor HD - datasheet*, (Online 18.4.2014)  
[http://www.wavetronix.com/uploads/download/file/419/smartsensor\\_hd\\_3datasheet-20131125142522.en.pdf](http://www.wavetronix.com/uploads/download/file/419/smartsensor_hd_3datasheet-20131125142522.en.pdf)